Volume 2, Issue 1, March 2022

INTELLIGENCE & ROBOTICS

Evolution of adaptive learning for nonlinear dynamic systems: a systematic survey

Mouhcine Harib, Hicham Chaoui, Suruz Miah

Open Access
ISSN 2770-3541 (Online)
www.intellrobot.com

Editor-in-Chief

Simon X. Yang

Prof. Simon X. Yang is currently the Head of the Advanced Robotics and Intelligent Systems Laboratory at the University of Guelph. His research interests include artificial intelligent, robotics, sensors and multi-sensor fusion, wireless sensor networks, control systems, bio-inspired intelligence, machine learning, neural networks, fuzzy systems, and computational neuroscience.

Our Features

(1) Gold Open Access
(2) Strong Editorial Board
(3) Rigorous Peer-review
(4) Free English language editing service
(5) Online first once accepted
(6) Free Publication before 31 Dec 2024
(7) Wide promotion (Twitter\LinkedIn\WeChat\Facebook)

Editorial Board

- 1 Editor-in-Chief
- 5 Advisory Editorial Members
- 24 Associate Editors
- 15 Youth Editorial Board Members

Scope

Top-quality unpublished original technical and non-technical application-focused articles are welcome from intelligence and robotics, particularly on the interdisciplinary areas of intelligence and robotics, including but not limited to the following areas:

- biological, bio-inspired, and artificial intelligence;
- neural networks, fuzzy systems, and evolutionary algorithms;
- sensing, multi-sensor fusion, localization, data analysis, modeling, planning, and control for various mobile, aerial, and underwater robotic systems;
- robot cooperation, teleoperation, and human-machine interactions.
- development and maintenance of real-world intelligent and robotic systems by multidisciplinary teams of scientists and engineers.

Journal Home
https://intellrobot.com/

Submission Link
https://oaemesas.com/login?JournalId=ir

EDITORIAL BOARD

GENERAL INFORMATION

About the Journal

Intelligence & Robotics (*IR*), ISSN 2770-3541 (Online), publishes top-quality unpublished original technical and non-technical application-focused articles on intelligence and robotics, particularly on the interdisciplinary areas of intelligence and robotics. The Journal seeks to publish articles that deal with the theory, design, and applications of intelligence and robotics, ranging from software to hardware. The scope of the Journal includes, but is not limited to, biological, bio-inspired, and artificial intelligence; neural networks, fuzzy systems, and evolutionary algorithms; sensing, multi-sensor fusion, localization, data analysis, modeling, planning, and control for various mobile, aerial, and underwater robotic systems; and robot cooperation, teleoperation and human-machine interactions. The Journal would be interested in distributing development and maintenance of real-world intelligent and robotic systems by multidisciplinary teams of scientists and engineers.

Information for Authors

Manuscripts should be prepared in accordance with Author Instructions.
Please check https://intellrobot.com/pages/view/author_instructions for details.
All manuscripts should be submitted online at https://oaemesas.com/login?JournalId=ir.

Copyright

Permissions

For information on how to request permissions to reproduce articles/information from this journal, please visit www.intellrobot.com.

Disclaimer

The information and opinions presented in the journal reflect the views of the authors and not of the journal or its Editorial Board or the Publisher. Publication does not constitute endorsement by the journal. Neither the *IR* nor its publishers nor anyone else involved in creating, producing or delivering the *IR* or the materials contained therein, assumes any liability or responsibility for the accuracy, completeness, or usefulness of any information provided in the *IR*, nor shall they be liable for any direct, indirect, incidental, special, consequential or punitive damages arising out of the use of the *IR*. *IR*, nor its publishers, nor any other party involved in the preparation of material contained in the *IR* represents or warrants that the information contained herein is in every respect accurate or complete, and they are not responsible for any errors or omissions or for the results obtained from the use of such material. Readers are encouraged to confirm the information contained herein with other sources.

Published by

OAE Publishing Inc.
245 E Main Street Ste 107, Alhambra CA 91801, USA
Website: www.oaepublish.com

Contacts

E-mail: editorial@intellrobot.com
Website: www.intellrobot.com

Intelligence & Robotics

CONTENTS

Volume 2, Issue 1

Research Article

Review

Research Article

Chazhoor *et al*. *Intell Robot* 2022;2:1-19
DOI: 10.20517/ir.2021.15

Intelligence & Robotics

Research Article

Open Access

Deep transfer learning benchmark for plastic waste classification

Anthony Ashwin Peter Chazhoor[1], Edmond S. L. Ho[1], Bin Gao[2], Wai Lok Woo[1]

[1]Department of Computer and Information Sciences, Northumbria University, Newcastle upon Tyne NE1 8ST, UK.
[2]School of Automation Engineering, University of Electronic Science and Technology of China, Chengdu 610000, Sichuan, China.

Correspondence to: Prof. Wai Lok Woo, Department of Computer and Information Sciences, Northumbria University, Ellison Place, Newcastle upon Tyne NE1 8ST, UK. E-mail: wailok.woo@northumbria.ac.uk

How to cite this article: Chazhoor AAP, Ho ESL, Gao B, Woo WL. Deep transfer learning benchmark for plastic waste classification. *Intell Robot* 2022;2:1-19. https://dx.doi.org/10.20517/ir.2021.15

Received: 2 Nov 2021 **First Decision:** 3 Dec 2021 **Revised:** 24 Dec 2021 **Accepted:** 18 Jan 2022 **Published:** 28 Jan 2022

Academic Editors: Simon X. Yang, Nallappan Gunasekaran **Copy Editor:** Xi-Jun Chen **Production Editor:** Xi-Jun Chen

Abstract

Millions of people throughout the world have been harmed by plastic pollution. There are microscopic pieces of plastic in the food we eat, the water we drink, and even the air we breathe. Every year, the average human consumes 74,000 microplastics, which has a significant impact on their health. This pollution must be addressed before it has a significant negative influence on the population. This research benchmarks six state-of-the-art convolutional neural network models pre-trained on the ImageNet Dataset. The models Resnet-50, ResNeXt, MobileNet_v2, DenseNet, SchuffleNet and AlexNet were tested and evaluated on the WaDaBa plastic dataset, to classify plastic types based on their resin codes by integrating the power of transfer learning. The accuracy and training time for each model has been compared in this research. Due to the imbalance in the data, the under-sampling approach has been used. The ResNeXt model attains the highest accuracy in fourteen minutes.

Keywords: Plastic, transfer learning, recycling, waste, classification

1. INTRODUCTION

Plastic finds itself in everyday human activities. The mass production of plastic was introduced in 1907 by Leo Baekeland, proved to be a boon to humankind[1]. Over the years, plastic has increasingly become an everyday necessity for humanity. The population explosion has a critical part in increasing domestic plastic usage[2]. Lightweight plastics have a crucial role in the transportation industry. Their usage in space exploration gives enormous leverage over heavy and expensive alternatives[3]. The packaging industry widely uses plastics after the e-commerce revolution because they are lightweight, cheap, and abundant. In 2015, the packing sector produced 141 million metric tons of garbage, accounting for 97 percent of all waste produced concerning the total consumption in the packaging sector[4]. Discarded polyethylene terephthalate (PETE) bottles are a common source of household waste. In 2021, global waste plastic bottle consumption will surpass 500 billion as estimated[2].

The increasing use of plastics and their wastage negatively affect the global economy. This surge in consumption and the low degradability of plastic have resulted in massive plastic accumulation in the environment, which has harmed ecosystems and human health[5]. This has resulted in countries formulating strict policies for plastics and even banning some types of single-use plastics. Plastics are non-biodegradable and considerably take a longer time to degrade. Reusing and recycling are viable ways to stop contaminating the environment with plastic pollution[6]. Plastic wastes can be retrieved after entering the municipal treatment plants or before it. However, the plastic waste from the municipal treatment plants is usually contaminated and ends up in landfills or incineration centers. The plastic waste collected outside of such plants is relatively cleaner and can be reused or recycled. Recovered plastics from such wastes have varied types of plastic, making it extremely difficult to identify and sort different kinds of plastics.

By integrating transfer learning, the Dataset needs only a limited number of input images to acquire high accuracy, and it also accelerates the training of neural networks, consequently improving the classification of multiple classes in a dataset[7]. Balancing the number of images in each class compensates for the class imbalance problem. This research contributes towards benchmarking of pre-trained models and concluding that the ResNeXt model achieves the highest accuracy on the WaDaBa dataset from the list of pre-trained models specified in this paper.

1.1. Literature review

Seven different varieties of plastics exist in the modern day. They are classified as Polyethylene terephthalate (PET or PETE), high-density polyethylene (HDPE), polyvinyl chloride (PVC or Vinyl), low-density polyethylene (LDPE), polypropylene (PP), polystyrene (PS or Styrofoam) and Others, which does not belong to any of the above types, has been shown in Figure 1[3].

1.1.1. Traditional sorting techniques

Initially, segregation of wastes and separation of different types of plastics were done manually. However, this results in increased labor costs and time consumption[6]. Traditional macro sorting of plastics was performed with the aid of sensors which included near-infrared spectrometers[8,9], x-ray transmission sensor, Fourier transformed Infrared Technique[10], laser aided identification, and marker identification by identifying the resin type[11]. However, these approaches are limited to recognizing just particular types of plastics and are costly due to the large equipment required. The intricacy of mechanical sorting and its maintenance, as well as the high initial investment, are the drawbacks of traditional sorting methods.

1.1.2. Modern sorting techniques

Deep learning has made classification easier, more efficient, and cost-effective, with less human intervention. The deep learning approach was enhanced by convolutional neural networks (CNN)[12]. CNNs are excellent for object classification and detection[13]. After the model has been trained on the data, the plastics may be sorted into the appropriate classes with the assistance of CNN. They do, however, require a huge quantity of training data, which might be difficult to get at times. When the input data is small, the problem of overfitting develops, resulting in inaccurate classifications[14]. Transfer learning reduces the

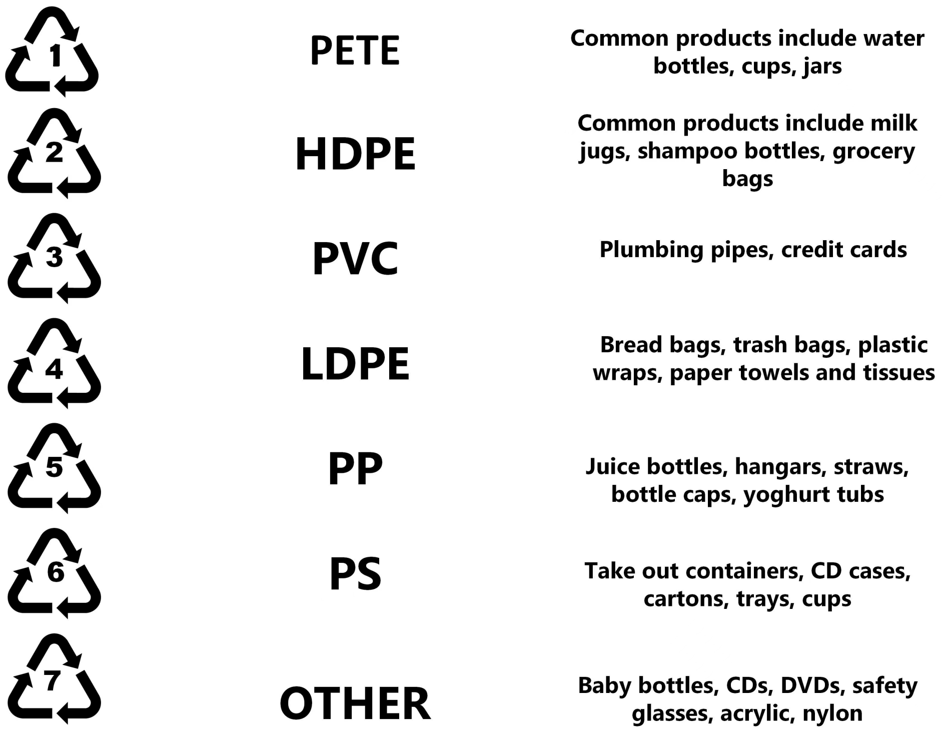

1
PETE
Common products include water bottles, cups, jars
2
HDPE
Common products include milk jugs, shampoo bottles, grocery bags
3
PVC
Plumbing pipes, credit cards
4
LDPE
Bread bags, trash bags, plastic wraps, paper towels and tissues
5
PP
Juice bottles, hangars, straws, bottle caps, yoghurt tubs
6
PS
Take out containers, CD cases, cartons, trays, cups
7
OTHER
Baby bottles, CDs, DVDs, safety glasses, acrylic, nylon

Figure 1. Types of plastic, its resin code and everyday examples of plastics. PETE: Polyethylene terephthalate; HDPE: high-density polyethylene; PVC: polyvinyl chloride; LDPE: low-density polyethylene; PP: polypropylene, PS: polystyrene.

training time of a CNN by pre-training the model using benchmark datasets such as ImageNet.

Bobulski *et al.*[15] proposed an end-to-end system with a micro-computer embedded with the vision to sort the PETE types of plastics in the WaDaBa dataset. The authors introduced data augmentation, which reduced the number of parameters but exponentially increased the number of samples, increasing the training time. Bobulski *et al.*[16] also proposed to classify distinct plastic categories based on a gradient feature vector. Agarwal *et al.*[17] presented Siamese and triplet loss neural networks to classify the WaDaBa dataset and succeeded with very high accuracy. However, this method requires a significant amount of time for training the neural networks. Chazhoor *et al.*[18] Anthony utilised transfer learning to compare the three most often used architectures (ResNeXt, Resnet-50-50 and AlexNet) on the WaDaBa dataset to select the optimal model; however, the K-fold cross validation technique was not applied; as a result, testing accuracy would vary widely.

The aim of the paper is to provide researchers with benchmark accuracies and the average time required to train on the WaDaBa dataset using the latest CNN models utilising cross-validation to categorise a range of plastics into their appropriate resin types. An unbiased and concrete set of parameters has been set to evaluate the Dataset to compare the models fairly[19]. This benchmark work will assist in gaining an impartial view of numerous recent CNN models applied to the WaDaBa dataset, establishing a baseline for future research. The models used in this paper are AlexNet[20], Resnet-50[21], ResNeXt[22], SqueezeNet[23], MobileNet_v2[24] and DenseNet[25].

2. METHODS

2.1. Dataset

The WaDaBa dataset is a sophisticated collection that contains images of common plastics used in society. The dataset includes seven distinct varieties of plastic. Images show several forms of plastics on a platform under two lighting conditions: an LED bulb and a fluorescent lamp and is displayed in Figure 2. Table 1 shows the distribution of the 4000 images in the dataset according to their classes. As there are no images in the PVC and PE-LD classes, both the classes have been excluded from the deep learning models. Deep learning models are trained on five class types with images in the current work i.e., PETE, PE-HD, PP, PS, and Other. The deep learning models are set up in such a way that each output matches one of the five class categories. When the images for PVC and PE-LD are released, these classes can be included in the models. The dataset's classes are imbalanced, with the last class holding just 40 images and the PETE class consisting of 2000 images. The dataset is freely accessible to the public[15].

2.2. Transfer learning

A large amount of data is needed to get optimum accuracy in a neural network. Data needs to be trained for hours on a powerful Graphical Processing Unit (GPU) to get the results. With the advent of transfer learning[26], there has been a significant change in the learning processes in deep neural networks. The model which has been already trained on a large dataset like ImageNet[27], known as the pre-trained model, enhances the transfer learning process. The transfer learning process works by freezing[28] the initially hidden layers of the model and fine-tuning the final layers of the models. The layer's frozen state indicates that it will not be trained. As a result, its weights will remain unchanged. As the data set used in this research is relatively small with a limited number of images in each class, transfer learning best suits this research. The pre-trained models used in the research are further explained in the subsection.

2.2.1. AlexNet

AlexNet is a neural network with three convolutional layers and two fully connected layers, and it was introduced in 2012 by Alex Krizhevesky. AlexNet increases learning capacity by increasing network depth and using multi-parameter tuning techniques. AlexNet uses ReLU to add non-linearity and dropout to decrease the overfitting of data. CNN-based applications gained popularity following AlexNet's excellent performance on the ImageNet dataset in 2012[23]. The architecture of AlexNet is shown in Figure 3.

2.2.2. Resnet-50

Residual networks (Resnet-50) are convolutional neural networks with skip connections with an extremely deep convolution and 11 million parameters. A skip connection after each block solves the vanishing gradient problem. The skip connection skips some layers in the network. With batch normalization and ReLU activation, two 3 × 3 convolutions are used in each block to achieve the desired result[21]. The architecture of Resnet-50-50 is displayed in Figure 4.

2.2.3. ResNeXt

Proposed by Facebook and ranking second in ILSVRC 2016, ResNeXt uses the repeating layer strategy of Resnet-5050, and it appends the split-transform-merge method[22]. The magnitude of a set of transformations is known as cardinality. Cardinality provides a novel approach to modifying model capacity by increasing the number of separate routes. Having width and depth as critical characteristics, ResNeXt adds on Cardinality as a new dimension. Increasing cardinality is a practical approach to enhance the accuracy of the model[22]. The architecture of ResNeXt is shown in Figure 5.

Table 1. The number of images corresponding to each class in the WaDaBa dataset[15]

Resin code	Class type	Number of images
1	PETE	2200
2	PE-HD	600
3	PVC	0
4	PE-LD	0
5	PP	640
6	PS	520
7	Other	40

PETE: Polyethylene terephthalate; PVC: polyvinyl chloride; PP: polypropylene, PS: polystyrene; PE-HD: high-density polyethylene; PE-LD: low-density polyethylene.

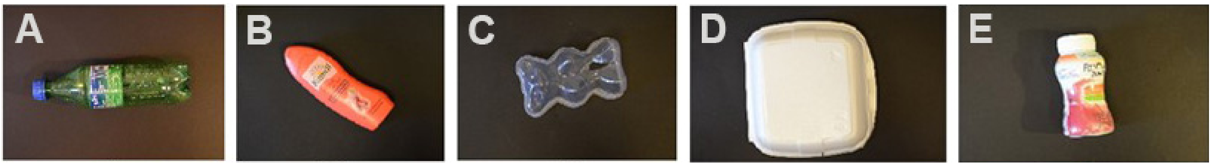


Figure 2. Examples of different types of plastics from the WaDaBa dataset in Figure 1. (A) Class 1 representing PETE (polyethylene terephthalate); (B) Class 2 representing HDPE (high-density polyethylene); (C) Class 5 representing PP (polypropylene); (D) Class 6 representing PS (polystyrene) ; (E) Class 7 representing Others[15].

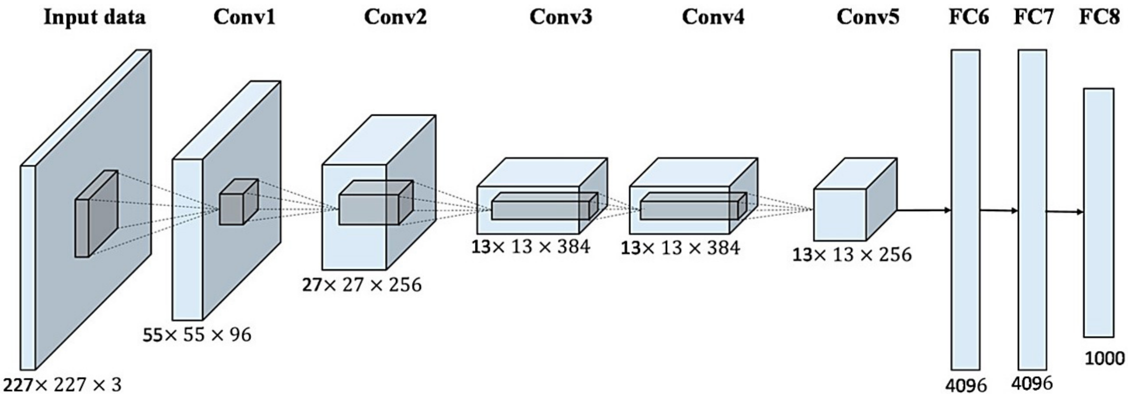


Figure 3. The architecture of AlexNet, having five convolutional layers and three fully connected layers. This figure is quoted with permission from Han *et al.*[29].

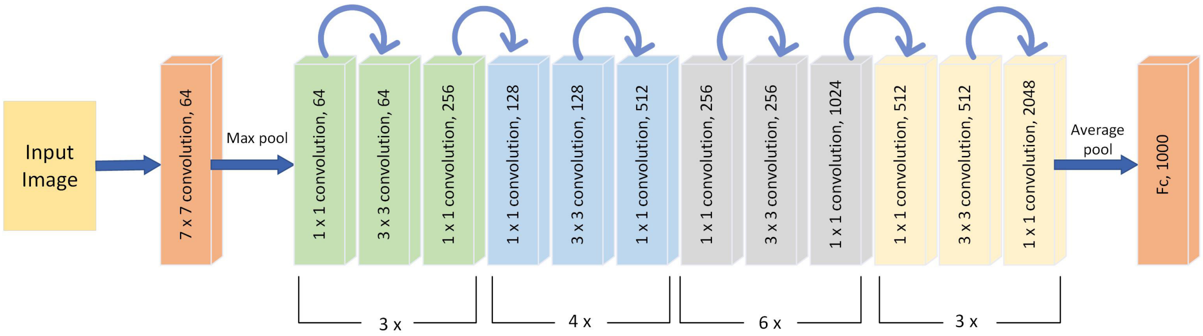


Figure 4. Architecture of Resnet-50-50. This figure is quoted with permission from Talo *et al.*[30].

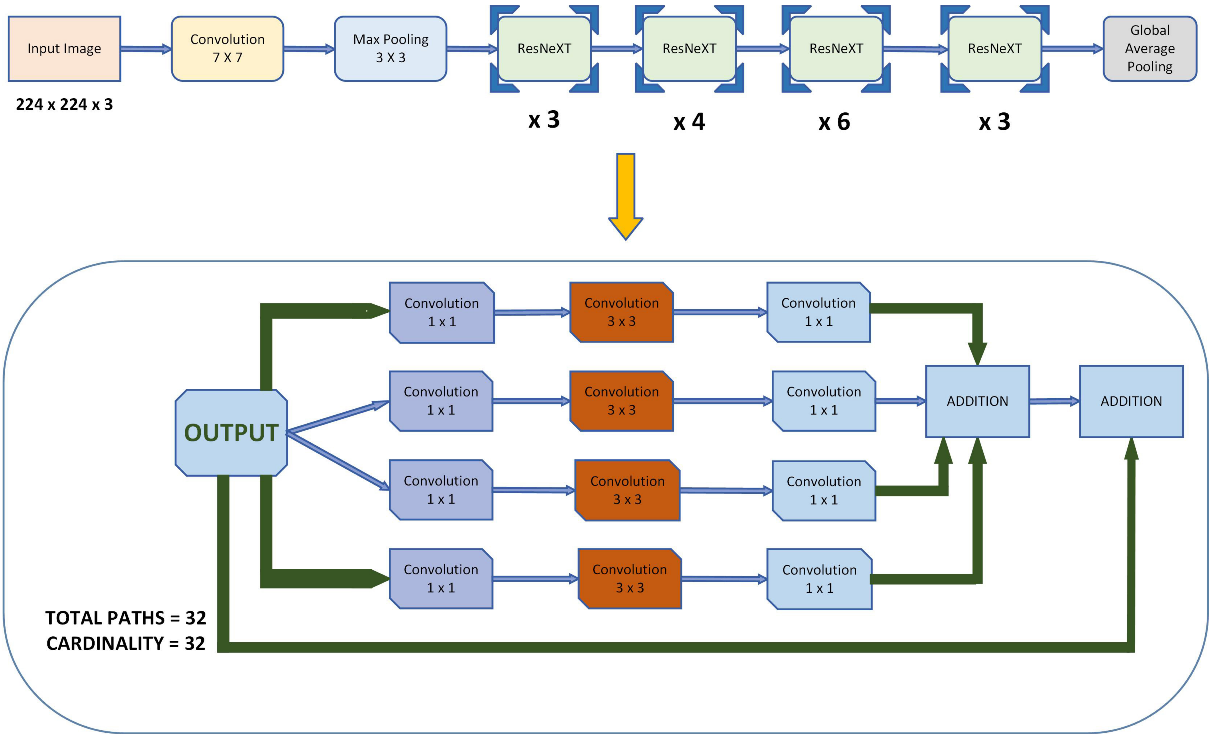


Figure 5. Architecture of ResNeXt. (Figure is redrawn and quoted from Go *et al.*[31])

2.2.4. MobileNet_v2

MobileNet_v2 is a CNN architecture built on an inverted residual structure, shortcut connections between narrow bottleneck layers to improve the mobile and embedded vision systems. A Bottleneck Residual Block is a type of residual block that creates a bottleneck using 1 × 1 convolutions. The number of parameters and matrix multiplications can be reduced by using a bottleneck. The goal is to make residual blocks as small as possible so that depth may be increased, and the parameters can be reduced. The model uses ReLU as the activation function. The architecture comprises a 32-filter convolutional layer at the top, followed by 19 bottleneck layers[24]. The architecture of MobileNet_v2 is shown in Figure 6.

2.2.5. DenseNet

Using a feed-forward system, DenseNet connects each layer to every other layer. Layers are created using feature maps from all previous levels, and their feature maps are utilized in all future layers to create new layers. They solve the vanishing-gradient problem and improve feature propagation and reuse while reducing the number of parameters significantly. The architecture of DenseNet is shown in Figure 7.

2.2.6. SqueezeNet

SqueezeNet is a small CNN that shrinks the network by reducing parameters while maintaining adequate accuracy. An entirely new building block has been introduced in the form of SqueezeNet's Fire module. A Fire module consists of a squeeze convolution layer containing only a 1 × 1 filter, which feeds into an expand layer having a combination of 1 × 1 and 3 × 3 convolution filters. Starting with an independent convolution layer, SqueezeNet then moves to 8 Fire modules before concluding with a final convolution layer. The architecture of SqueezeNet is shown in Figure 8.

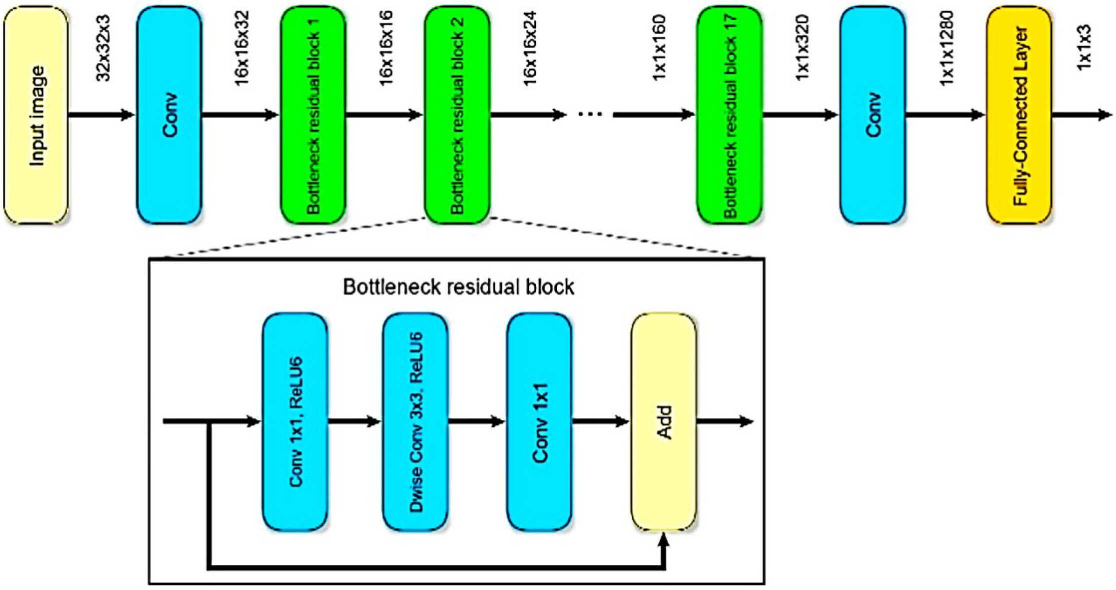


Figure 6. The architecture of MobileNet_v2. This figure is quoted with permission from Seidaliyeva *et al.*[32]

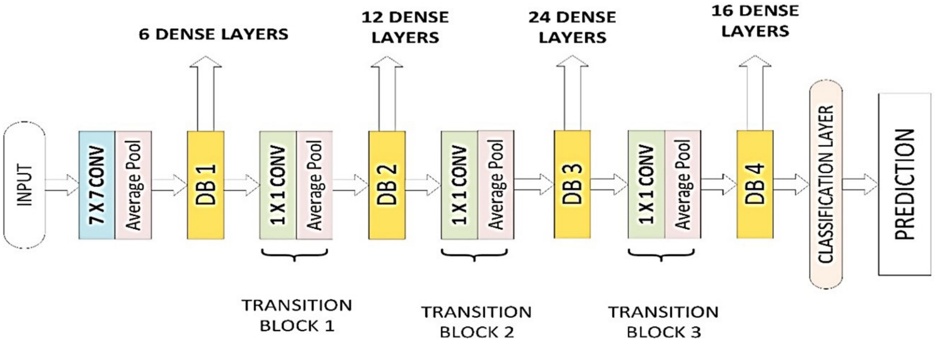


Figure 7. The architecture of DenseNet. This figure is quoted with permission from Huang *et al.*[25].

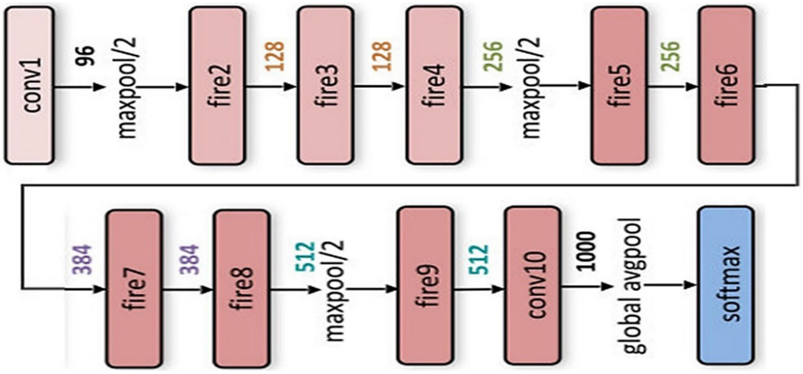


Figure 8. The architecture of SqueezeNet. This figure is quoted with permission from Nguyen *et al.*[33].

2.3. Experimental settings and the experiment

All the experiments were run on Ubuntu Linux operating system. The models were trained on Intel i7, 3.60 GHz, 32 GB ram and the graphical processing unit used was the Nvidia GeForce RTX 2080 Super. The deep learning framework used in this research is PyTorch[34]. The images from the WaDaBa dataset are input to the pre-trained models after performing under-sampling in the dataset. The batch size chosen for this experiment is 4 such that the GPU doesn't run out of memory while processing. The learning rate is 0.001 and is decayed by a factor of 0.1 every seven epochs. Decaying the learning rate aids the network's convergence to a local minimum and also enhances the learning of complicated patterns[35]. Cross-Entropy loss is utilized for training, accompanied by a momentum of 0.9, which is widely used in the machine learning and neural network communities[36]. The Stochastic Gradient Descent (SGD) optimizer[37], a gradient descent technique that is extensively employed in training deep learning models, is used. The training is done using a five-fold cross-validation technique, and the result is generated, along with graphs showing the number of epochs *vs.* accuracy and number of epochs *vs.* loss. On the WaDaBa dataset, each model was subjected to twenty epochs.

Before being forwarded on to the training, the data was normalized. These approaches, which were applied to the data, included random horizontal flipping and centre cropping.

The size of the input picture is 224 × 224 pixels [Figure 9].

2.3.1. Imbalance in the dataset

The number of images for each class in the dataset is uneven. The first class (PETE) contains 2200 photos, while the last class (Others) contains only 40. Due to the size and cost of certain forms of plastic, obtaining datasets is quite tricky. Because of the class imbalance, the under-sampling strategy was used. Images were split into training and validation sets, eighty percent for the training and twenty percent for the testing purposes.

2.3.2. K-fold cross-validation

The 5-fold cross-validation was considered for all the tests to validate the benchmark models[38]. The data was tested on the six models and the training loss and accuracy, validation loss and accuracy and the training time was recorded for 20 epochs with identical model parameters. The resultant average data was tabulated, and the corresponding graphs were plotted for visual representation. The flow chart of the experimental process is displayed in Figure 8.

3. RESULTS

3.1 Accuracy, loss, area under curve and receiver operating characteristic curve

The metrics used to benchmark the models on the WaDaBa dataset are accuracy and loss. The accuracy corresponds to the correctness of the value[39]. It measures the value to the actual value. Loss is a prediction of how erroneous the predictions of a neural network are, and the loss is calculated with the help of a loss function[40]. The area under curve (AUC) measures the classifier's ability to differentiate between classes and summarize the receiver operating characteristic (ROC) curve. ROC plots the performance of a classification model's overall accuracy. The curve plots the True Positive Rate against the False Positive Rate.

Table 2 clearly shows that the ResNeXt architecture achieves the maximum accuracy of 87.44 percent in an average time of thirteen minutes and eleven seconds. When implemented in smaller and portable devices, smaller networks such as MobileNet_v2, SqueezeNet, and DenseNet offer equivalent accuracy. AlexNet trains the model in the shortest period but with the lowest accuracy. In comparison to the other models, DenseNet takes the longest to train. With a classification accuracy of 97.6 percent, ResNeXt comes out as the top model for reliably classifying PE-HD. When compared to other models, MobileNet_v2 classifies PS with more accuracy. Also, from Table 2, we can see that PP has the least classification accuracy for all the

Table 2. The mean and class wise accuracies of the models pretrained on the ImageNet dataset, along with the time taken for training for 20 epochs. The standard deviation indicates the average deviation in accuracy across the five-folds in the respective model along with the total number of parameters for each model

	AlexNet	Resnet-50	ResNeXt	MoblineNet_v2	DenseNet	SqueezeNet
Mean accuracy (%)	80.08	85.54	**87.44**	87.35	85.58	82.59
PETE (%)	84.8	85	85	85	**88.8**	84.4
PE-HD (%)	85.0	95.4	**97.6**	94.2	95.6	91.4
PP (%)	67.2	68.6	74	**74.8**	66.4	66.8
PS (%)	80.2	86.0	83.2	**89.6**	85.4	82.2
Other (%)	100	100	100	100	100	97.5
Time (min)	11.8	12.05	13.11	12.06	17.33	12.01
Std. deviation σ (%)	7.5	4.9	5.4	6.0	5.3	1.7
No. of parameters (in million)	57	23	22	2	6	0.7

PETE: Polyethylene terephthalate; PP: polypropylene, PS: polystyrene.

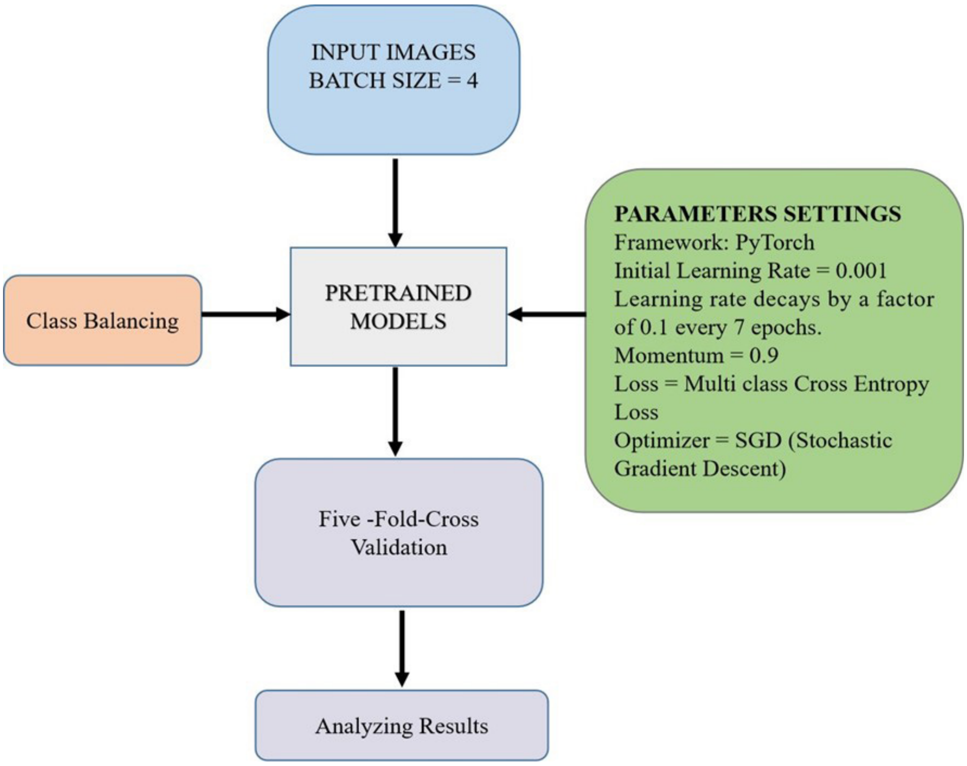

INPUT IMAGES
BATCH SIZE = 4
Class Balancing
PRETRAINED MODELS
PARAMETERS SETTINGS
Framework: PyTorch
Initial Learning Rate = 0.001
Learning rate decays by a factor of 0.1 every 7 epochs.
Momentum = 0.9
Loss = Multi class Cross Entropy Loss
Optimizer = SGD (Stochastic Gradient Descent)
Five -Fold-Cross Validation
Analyzing Results

Figure 9. Flowchart summarizing the experiment.

models. In Table 2, the standard deviation, σ, is displayed, which is a measure of how far values deviate from the mean. The standard deviation is given by the following unbiased estimation:

$$\sigma = \sqrt{\frac{\sum(x_i - \bar{x})^2}{n-1}}$$

x_i = accuracy at the i^{th} epoch

$\bar{x}$ = mean of the accuracies

n = total number of epochs (e.g., 20)

4. DISCUSSION

In the results section from Table 2, we can observe that ResNeXt architecture performs better than all the other architectures discussed in this paper. MobileNet_v2 architecture falls behind ResNeXt architecture with 0.1 % accuracy. Considering the time factor, MobileNet_v2 trains faster than ResNext by a minute's advantage. When the data is considerably large, the difference in time factor will increase, giving the MobileNet_v2 architecture dominance.

The validation loss of AlexNet architecture from Table 3 and SqueezeNet architecture from Table 4 does not significantly drop compared to other models used in the research and from the graph, it can be observed from Figure 10 and Figure 11 that there is a diverging gap between its accuracy loss and validation loss curves for both models. Fewer images in the Dataset and multiple classes cause this effect on the AlexNet architecture. Similar results can be observed for SqueezeNet from Table 4 and Figure 11, which have a similar architecture to AlexNet. Table 5 and Figure 12 represent the training and validation accuracies and loss values and their corresponding graphs for the pre-trained Resnet-50 model. From Table 6 and Figure 13, we can observe the training and validation accuracy and loss values and their plots for ResNeXt architecture. Similarly, from Table 7 and Figure 14, the accuracies and their graphs for MobileNet_v2 can be observed. The DenseNet architecture represented in Table 8 and Figure 15 takes the longest time to train and has a good accuracy score of 85.58%, which is comparable to the Resnet-50 architecture, having an accuracy of 85.54%. The five-fold cross-validation approach tests every data point in the dataset and helps improve the overall accuracy.

Figure 16 shows the AUC and ROC for all the models in this paper. The SqueezeNet and AlexNet architecture display the lowest AUC score. MobileNet_v2, Resnet-50, ResNext and DenseNet have a comparable AUC score. From the ROC curve, it can be inferred that the models can correctly distinguish between the types of plastics in the Dataset. ResNeXt architecture achieves the largest AUC.

5. CONCLUSION

When we compare our findings to previous studies in the field, we find that including transfer learning reduces total training time significantly. It will be simple to train the existing model and attain improved accuracy in a short amount of time if the WaDaBa dataset is enlarged in the future. This paper has benchmarked six state-of-the-art models on the WaDaBa plastic dataset by integrating deep transfer learning. This work will be laid out as a baseline work for future developments on the WaDaBa dataset. The paper focuses on supervised learning for plastic waste classification. Unsupervised learning procedures are one area where the article has placed less focus. The latter might be beneficial for pre-training or enhancing the supervised classification models using pre-trained feature selection. Pattern decomposition methods[41] like nonnegative matrix factorization[42] and ensemble joint sparse low rank matrix decomposition[43] are

Table 3. The mean training and validation accuracies and losses for AlexNet architecture for 20 epochs

Epoch	Mean_AlexNet			
	Training accuracy	Validation accuracy	Training loss	Validation loss
1	0.5815	0.57302	1.00228	1.1308
2	0.6675	0.64806	0.80658	1.09448
3	0.7177	0.5804	0.69244	1.1246
4	0.73384	0.64656	0.6721	1.01474
5	0.77882	0.67598	0.55144	0.9506
6	0.78652	0.66568	0.51194	1.04706
7	0.79548	0.7093	0.50188	0.84044
8	0.84654	0.7696	0.36054	0.82302
9	0.87302	0.7642	0.30162	0.89168
10	0.87962	0.77646	0.28896	0.90384
11	0.87458	0.77746	0.29108	0.92258
12	0.88206	0.78874	0.28282	0.8886
13	0.88462	0.78236	0.26542	0.99196
14	0.88192	0.78532	0.26406	0.99434
15	0.89248	0.78972	0.25636	0.98168
16	0.89126	0.78972	0.2576	0.98266
17	0.88914	0.79118	0.25864	0.95596
18	0.897	0.79608	0.24166	0.95004
19	0.89344	0.79706	0.24634	0.9735
20	0.89602	0.79414	0.24826	0.98582

Table 4. The mean training and validation accuracies and losses for SqueezeNet architecture for 20 epochs

Epoch	Mean SqueezeNet			
	Training accuracy	Validation accuracy	Training loss	Validation loss
1	0.47992	0.7281	1.02608	1.32476
2	0.64688	0.7437	0.78012	0.96076
3	0.7134	0.718	0.68612	1.05972
4	0.74428	0.67796	0.6426	1.14184
5	0.76116	0.7003	0.5903	0.81164
6	0.79006	0.70916	0.53186	0.88014
7	0.81026	0.65862	0.51222	0.89182
8	0.85586	0.69658	0.42766	0.81594
9	0.87364	0.70138	0.3871	0.89832
10	0.87874	0.70724	0.37834	0.99886
11	0.88684	0.6838	0.3752	0.9401
12	0.89062	0.69988	0.36256	0.93402
13	0.89798	0.69218	0.3465	0.94986
14	0.88878	0.7183	0.36842	0.8951
15	0.89504	0.70776	0.35906	0.97796
16	0.89798	0.70376	0.35146	1.0066
17	0.89896	0.70712	0.35242	0.99574
18	0.90166	0.70396	0.34732	1.00284
19	0.90422	0.70202	0.34508	1.01182
20	0.90238	0.70606	0.34562	0.9707

Table 5. The mean training and validation accuracies and losses for Resnet-50 architecture for 20 epochs

Epoch	Mean Resnet-50 values			
	Training accuracy	Validation accuracy	Training loss	Validation loss
1	0.5515	0.6706	1.12794	1.04068
2	0.69346	0.70782	0.81024	0.96718
3	0.7455	0.7691	0.66772	0.86036
4	0.77918	0.76568	0.5758	0.82058
5	0.80062	0.77648	0.52012	0.66052
6	0.8256	0.75932	0.44886	0.85278
7	0.83992	0.74364	0.42794	1.16314
8	0.87704	0.82598	0.32214	0.60218
9	0.89198	0.82254	0.2835	0.6571
10	0.90986	0.82942	0.24506	0.62152
11	0.90324	0.83382	0.2566	0.58042
12	0.91498	0.83234	0.23156	0.63032
13	0.91182	0.81626	0.23618	0.6429
14	0.91476	0.83726	0.23086	0.65462
15	0.9151	0.83484	0.2235	0.6636
16	0.91464	0.82894	0.22348	0.70444
17	0.91684	0.8343	0.21748	0.65494
18	0.91684	0.83776	0.21546	0.6189
19	0.91708	0.83482	0.22578	0.68982
20	0.91352	0.83922	0.22412	0.61236

Table 6. The mean training and validation accuracies and losses for ResNeXt architecture for 20 epochs

Epoch	Mean ResNeXt values			
	Training accuracy	Validation accuracy	Training loss	Validation loss
1	0.57454	0.71078	1.09714	0.97576
2	0.69518	0.74312	0.8304	0.87308
3	0.752	0.67498	0.66784	1.3998
4	0.79228	0.76764	0.57174	0.93114
5	0.81336	0.78234	0.52164	0.7225
6	0.83306	0.83136	0.4542	0.70478
7	0.84494	0.81374	0.42144	0.7807
8	0.88366	0.8564	0.30548	0.5644
9	0.89836	0.85442	0.28038	0.64594
10	0.90642	0.85294	0.26156	0.62974
11	0.90826	0.85834	0.2503	0.65006
12	0.9145	0.85	0.2385	0.6518
13	0.9084	0.84118	0.2411	0.64972
14	0.91084	0.8544	0.24424	0.59668
15	0.91316	0.85246	0.2417	0.55656
16	0.92564	0.84854	0.2097	0.58186
17	0.91156	0.85882	0.23282	0.58778
18	0.916	0.85688	0.22358	0.63122
19	0.91598	0.84658	0.223	0.62936
20	0.92014	0.85246	0.21606	0.65276

Table 7. The mean training and validation accuracies and losses for MobileNet_v2 architecture for 20 epochs

Epoch	Mean MobileNet_v2			
	Training accuracy	**Validation accuracy**	**Training loss**	**Validation loss**
1	0.55528	0.66322	1.12416	0.97572
2	0.64264	0.71714	0.94286	0.79604
3	0.6871	0.77108	0.806	0.77816
4	0.72912	0.7392	0.70786	0.89686
5	0.75566	0.74462	0.6542	0.8389
6	0.7858	0.78334	0.57576	0.75382
7	0.78846	0.7799	0.54498	0.86344
8	0.8392	0.83332	0.4141	0.62084
9	0.85942	0.8495	0.36976	0.57796
10	0.8649	0.85296	0.35118	0.57304
11	0.87458	0.84954	0.33336	0.57328
12	0.87606	0.85734	0.32184	0.5281
13	0.8768	0.86618	0.3207	0.50986
14	0.88106	0.84902	0.31194	0.545
15	0.88464	0.85344	0.30746	0.53638
16	0.88756	0.86178	0.2966	0.5141
17	0.88804	0.8613	0.30038	0.50172
18	0.88342	0.8608	0.30566	0.52828
19	0.88512	0.85688	0.30972	0.53054
20	0.8822	0.86176	0.31576	0.50632

Table 8. The mean training and validation accuracies and losses for DenseNet architecture for 20 epochs

Epoch	Mean DenseNet			
	Training accuracy	**Validation accuracy**	**Training loss**	**Validation loss**
1	0.55724	0.6446	1.0884	1.04494
2	0.68426	0.73088	0.81858	0.74552
3	0.7488	0.72302	0.6718	1.14064
4	0.76168	0.75196	0.64602	0.90288
5	0.7874	0.79118	0.5675	0.69646
6	0.81936	0.76862	0.50594	0.85718
7	0.82216	0.77744	0.48568	0.76844
8	0.87188	0.79952	0.36034	0.66998
9	0.87814	0.83136	0.31836	0.51186
10	0.8911	0.80736	0.30766	0.5814
11	0.8954	0.82354	0.28282	0.58526
12	0.90164	0.83874	0.27306	0.59644
13	0.89908	0.8392	0.2748	0.5592
14	0.9019	0.84118	0.27446	0.57224
15	0.90704	0.83578	0.25116	0.5755
16	0.9096	0.84366	0.24786	0.5398
17	0.90582	0.84216	0.24938	0.5301
18	0.9063	0.84316	0.26094	0.60658
19	0.91196	0.8299	0.24698	0.57962
20	0.9079	0.84364	0.24388	0.52476

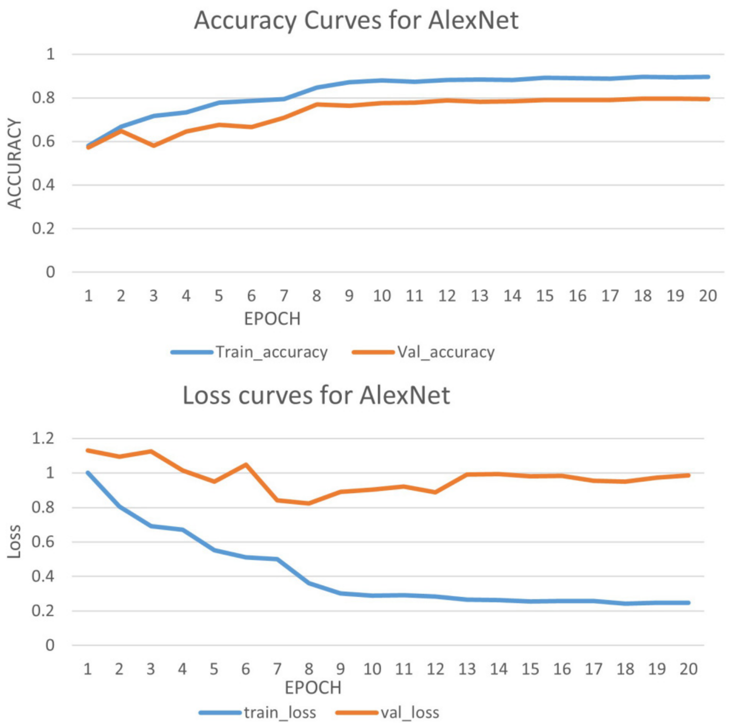


Figure 10. Accuracy and loss curves for AlexNet architecture.

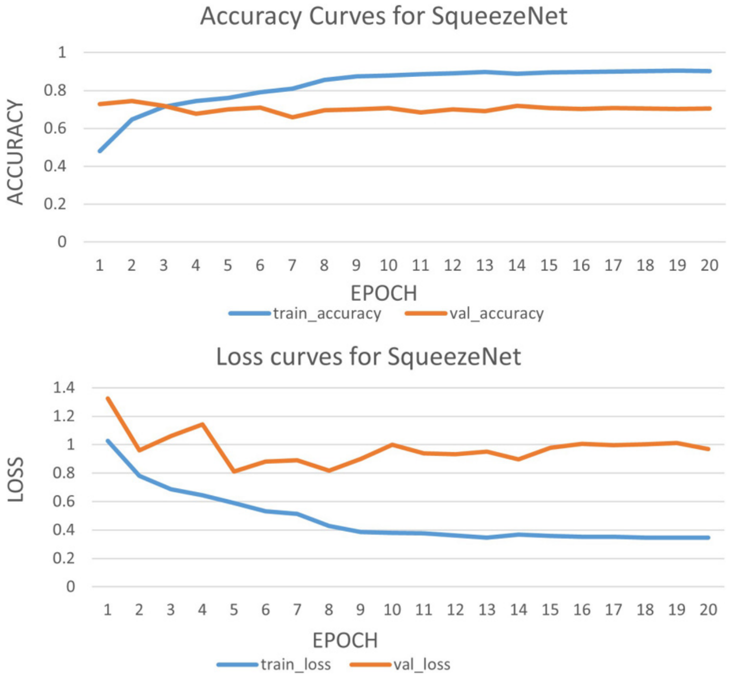


Figure 11. Accuracy and loss curves for SqueezeNet architecture.

examples of unsupervised learning strategies. Higher order decomposition approaches, such as low-rank tensor decomposition[44,45] and hierarchical sparse tensor decomposition[46], can result in improved performance. This would be the future path of study to improve plastic waste classification.

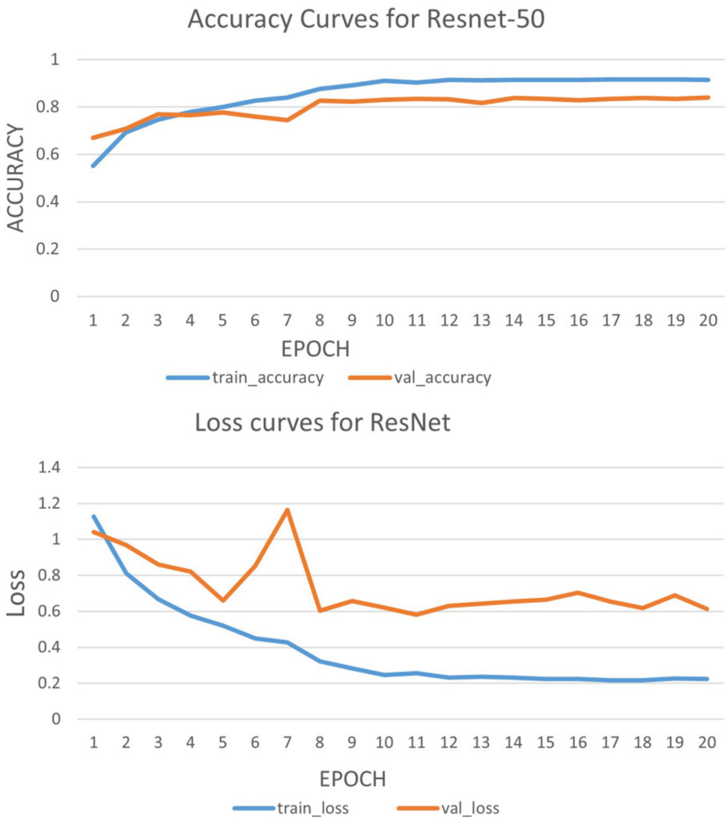


Figure 12. Accuracy and loss curves for Resnet-50 architecture.

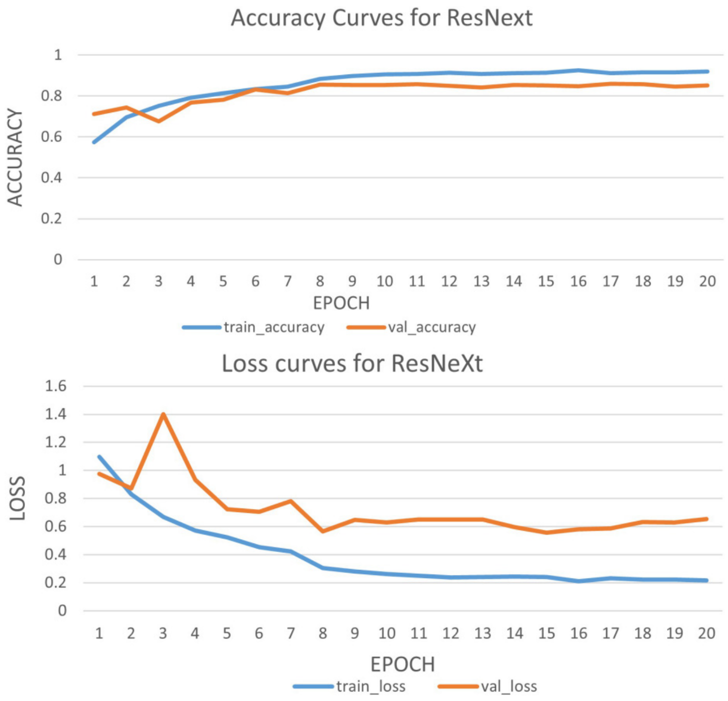


Figure 13. Accuracy and loss curves for ResNeXt architecture.

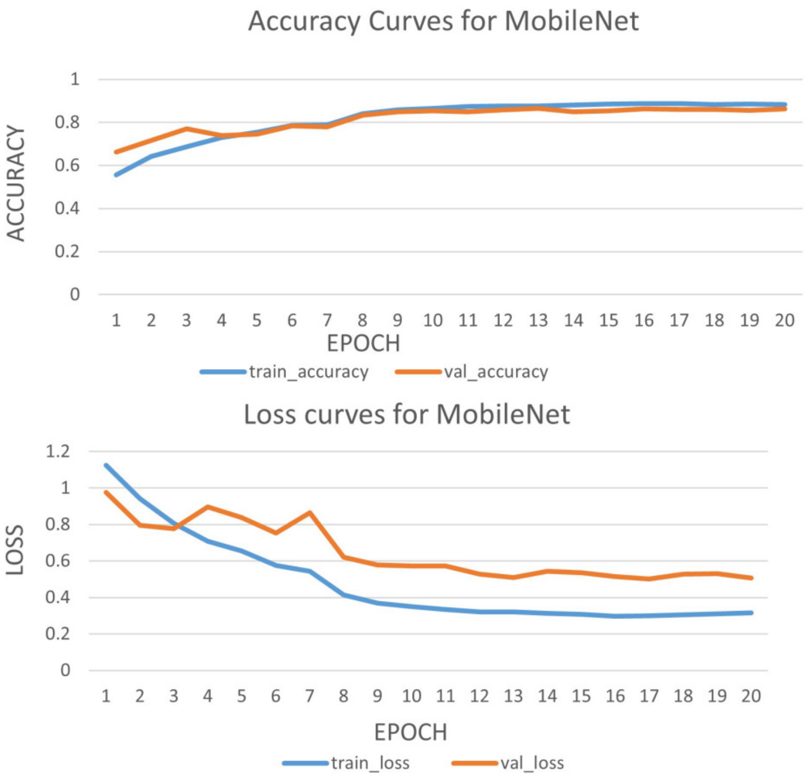


Figure 14. Accuracy and loss curves for MobileNet_v2 architecture.

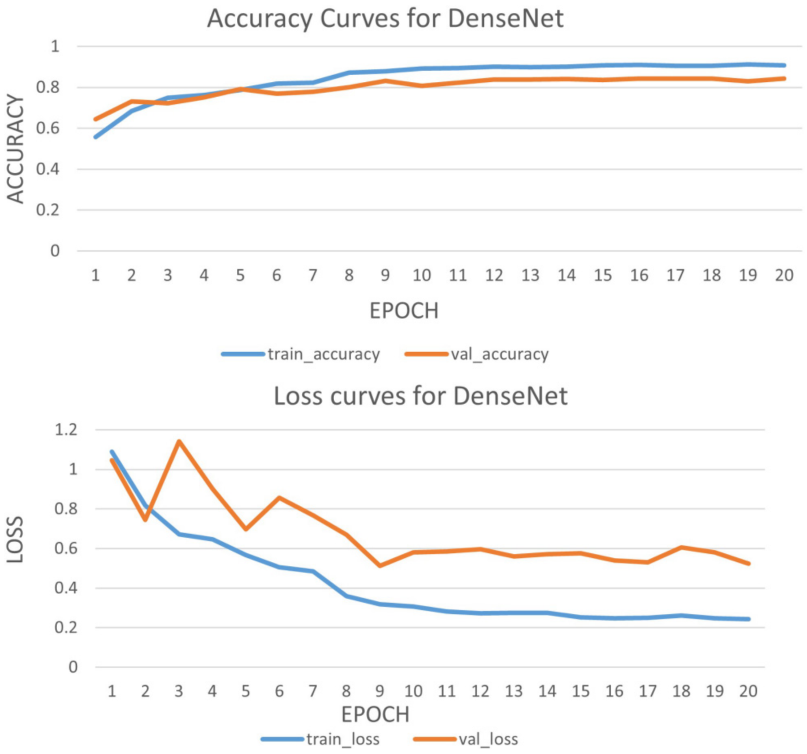


Figure 15. Accuracy and loss curves for DenseNet architecture.

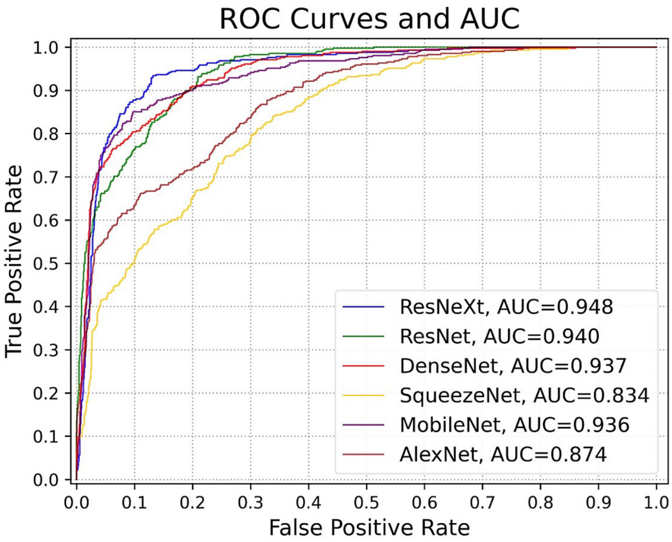


Figure 16. Area under curve and receiver operating characteristic for Resnet-50, ResNeXt, DenseNet, SqueezeNet, MobileNet_v2 and AlexNet models. AUC: Area under curve; ROC: receiver operating characteristic.

DECLARATIONS

Authors' contributions

Investigated the research area, reviewed and summarized the literature, wrote and edited the original draft: Chazhoor AAP

Managed the research activity planning and execution, contributed to the development of ideas according to the research aims: Ho ESL

Performed critical review, commentary and revision, funding acquisition: Gao B

Managed the research activity planning and execution, contributed to the development of ideas according to the research aims, funding acquisition, provided administrative: Woo WL

Availability of data and materials

The data can be found at http://wadaba.pcz.pl/. Emailing the creator by signing a consent form will give password access to the data[15]. The code has been uploaded to GitHub and the link is: https://github.com/ashys2012/plastic_wadaba/tree/main.

Financial support and sponsorship

The project is partially funded by Northumbria University and National Natural Science Foundation of China (No. 61527803, No. 61960206010).

Conflicts of interest

All authors declared that there are no conflicts of interest.

Ethical approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

Copyright

REFERENCES

1. Hiraga K, Taniguchi I, Yoshida S, Kimura Y, Oda K. Biodegradation of waste PET: a sustainable solution for dealing with plastic pollution. *EMBO Rep* 2019;20:e49365. DOI PubMed PMC
2. Alqattaf A. Plastic waste management: global facts, challenges and solutions. 2020 Second International Sustainability and Resilience Conference: Technology and Innovation in Building Designs(51154). 2020 Nov 11-12; Sakheer, Bahrain. IEEE; 2020. p. 1-7. DOI
3. Klemeš JJ, Fan YV. Plastic replacements: win or loss? 2020 5th International Conference on Smart and Sustainable Technologies (SpliTech). 2020 Sep 23-26; Split, Croatia. IEEE; 2020. p. 1-6. DOI
4. Backstrom J, Kumar N. Advancing the circular economy of plastics through eCommerce. Available from: https://hdl.handle.net/1721.1/130968 [Last accessed on 24 Jan 2022].
5. Joshi C, Browning S, Seay J. Combating plastic waste via Trash to Tank. *Nat Rev Earth Environ* 2020;1:142-142. DOI
6. Siddique R, Khatib J, Kaur I. Use of recycled plastic in concrete: a review. *Waste Manag* 2008;28:1835-52. DOI PubMed
7. Jiao W, Wang Q, Cheng Y, Zhang Y. End-to-end prediction of weld penetration: a deep learning and transfer learning based method. *J Manuf Process* 2021;63:191-7. DOI
8. Duan Q, Li J. Classification of common household plastic wastes combining multiple methods based on near-infrared spectroscopy. *ACS EST Eng* 2021;1:1065-73. DOI
9. Masoumi H, Safavi SM, Khani Z. Identification and classification of plastic resins using near infrared reflectance. *Int J Mech Ind Eng* 2012;6:213-20. DOI
10. Veerasingam S, Ranjani M, Venkatachalapathy R, et al. Contributions of Fourier transform infrared spectroscopy in microplastic pollution research: a review. *Crit Rev Environ Sci Technol* 2021;51:2681-743. DOI
11. Bruno EA. Automated sorting of plastics for recycling. Available from: https://www.semanticscholar.org/paper/Automated-Sorting-of-Plastics-for-Recycling-Edward-Bruno/e6e5110c06f67171409bab3b38f742db6dc110fc [Last accessed on 24 Jan 2022].
12. Alzubaidi L, Zhang J, Humaidi AJ, et al. Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. *J Big Data* 2021;8:53. DOI PubMed PMC
13. Albawi S, Mohammed TA, Al-Zawi S. Understanding of a convolutional neural network. 2017 International Conference on Engineering and Technology (ICET). 2017 Aug 21-23; Antalya, Turkey. IEEE;2017. p. 1-6. DOI
14. Xie L, Wang J, Wei Z, Wang M, Tian Q. Disturblabel: regularizing CNN on the loss layer. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2016 Jun 27-30; Las Vegas, NV, USA. IEEE; 2016. p. 4753-62. DOI
15. Bobulski J, Piatkowski J. PET waste classification method and plastic waste DataBase - WaDaBa. In: Choraś M, Choraś RS, editors. Image processing and communications challenges 9. Cham: Springer International Publishing; 2018. p. 57-64. DOI
16. Bobulski J, Kubanek M. Waste classification system using image processing and convolutional neural networks. In: Rojas I, Joya G, Catala A, editors. Advances in computational intelligence. Cham: Springer International Publishing; 2019. p. 350-61. DOI
17. Agarwal S, Gudi R, Saxena P. One-Shot learning based classification for segregation of plastic waste. 2020 Digital Image Computing: Techniques and Applications (DICTA). 2020 Nov 29-2020 Dec 2; Melbourne, Australia. IEEE; 2020. p. 1-3. DOI
18. Chazhoor AAP, Zhu M, Ho ES, Gao B, Woo WL. Intelligent classification of different types of plastics using deep transfer learning. Available from: https://researchportal.northumbria.ac.uk/ws/portalfiles/portal/55869518/ROBOVIS_2021_33_CR.pdf [Last accessed on 24 Jan 2022].
19. Guo Y, Zhang L, Hu Y, He X, Gao J. MS-Celeb-1M: a dataset and benchmark for large-scale face recognition. In: Leibe B, Matas J, Sebe N, Welling M, editors. Computer Vision - ECCV 2016. Cham: Springer International Publishing; 2016. p. 87-102. DOI
20. Krizhevsky A, Sutskever I, Hinton GE. Imagenet classification with deep convolutional neural networks. *Adv Neural Inf Process Syst* 2012;25:1097-105. DOI
21. He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2016 Jun 27-30; Las Vegas, NV, USA. IEEE; 2016. p. 770-8. DOI
22. Xie S, Girshick R, Dollár P, Tu Z, He K. Aggregated residual transformations for deep neural networks. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2017 Jul 21-26; Honolulu, HI, USA. IEEE; 2017. p. 5987-95. DOI
23. Iandola FN, Han S, Moskewicz MW, Ashraf K, Dally WJ, Keutzer K. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and< 0.5 MB model size. Available from: https://arxiv.org/abs/1602.07360 [Last accessed on 24 Jan 2022].
24. Sandler M, Howard A, Zhu M, Zhmoginov A, Chen L-C. Mobilenetv2: Inverted residuals and linear bottlenecks. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2018 Jun 18-23; Salt Lake City, UT, USA. IEEE; 2018. p. 4510-20. DOI
25. Huang G, Liu Z, Van Der Maaten L, Weinberger KQ. Densely connected convolutional networks. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2017 Jul 21-26; Honolulu, HI, USA. IEEE; 2017. p. 2261-9. DOI
26. Tan C, Sun F, Kong T, Zhang W, Yang C, Liu C. A survey on deep transfer learning. In: Kůrková V, Manolopoulos Y, Hammer B, Iliadis L, Maglogiannis I, editors. Artificial neural networks and machine learning - ICANN 2018. Cham: Springer International Publishing; 2018. p. 270-9. DOI
27. Deng J, Dong W, Socher R, Li L-J, Li K, Fei-Fei L. Imagenet: a large-scale hierarchical image database. 2009 IEEE Conference on Computer Vision and Pattern Recognition. 2009 Jun 20-25; Miami, FL, USA. IEEE; 2009. p. 248-55. DOI
28. Brock A, Lim T, Ritchie JM, Weston N. Freezeout: accelerate training by progressively freezing layers. Available from: https://arxiv.org/abs/1706.04983 [Last accessed on 24 Jan 2022].

29. Han X, Zhong Y, Cao L, Zhang L. Pre-trained AlexNet Architecture with pyramid pooling and supervision for high spatial resolution remote sensing image scene classification. *Remote Sensing* 2017;9:848. DOI
30. Talo M. Convolutional neural networks for multi-class histopathology image classification. 2019. Available from: https://arxiv.org/ftp/arxiv/papers/1903/1903.10035.pdf [Last accessed on 24 Jan 2022].
31. Go JH, Jan T, Mohanty M, Patel OP, Puthal D, Prasad M. Visualization approach for malware classification with ResNeXt. 2020 IEEE Congress on Evolutionary Computation (CEC). 2020 Jul 19-24; Glasgow, UK. IEEE; 2020. p. 1-7. DOI
32. Seidaliyeva U, Akhmetov D, Ilipbayeva L, Matson ET. Real-time and accurate drone detection in a video with a static background. *Sensors (Basel)* 2020;20:3856. DOI PubMed PMC
33. Nguyen THB, Park E, Cui X, Nguyen VH, Kim H. fPADnet: small and efficient convolutional neural network for presentation attack detection. *Sensors (Basel)* 2018;18:2532. DOI PubMed PMC
34. Paszke A, Gross S, Chintala S, et al. Automatic differentiation in pytorch. Available from: https://openreview.net/pdf?id=BJJsrmfCZ [Last accessed on 24 Jan 2022].
35. You K, Long M, Wang J, Jordan MI. How does learning rate decay help modern neural networks? Available from: https://arxiv.org/abs/1908.01878 [Last accessed on 24 Jan 2022].
36. Li X, Chang D, Tian T, Cao J. Large-margin regularized Softmax cross-entropy loss. *IEEE Access* 2019;7:19572-8. DOI
37. Ketkar N. Stochastic gradient descent. Deep learning with Python. Springer; 2017. p. 113-32. DOI
38. Mukherjee H, Ghosh S, Dhar A, Obaidullah SM, Santosh KC, Roy K. Shallow convolutional neural network for COVID-19 outbreak screening using chest X-rays. *Cognit Comput* 2021. DOI PubMed PMC
39. Selvik JT, Abrahamsen EB. On the meaning of accuracy and precision in a risk analysis context. *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability* 2017;231:91-100. DOI
40. Singh A, Príncipe JC. A loss function for classification based on a robust similarity metric. The 2010 International Joint Conference on Neural Networks (IJCNN). 2010 Jul 18-23; Barcelona, Spain. IEEE; 2010. p. 1-6. DOI
41. Gao B, Bai L, Woo WL, Tian G. Thermography pattern analysis and separation. *Appl Phys Lett* 2014;104:251902. DOI
42. Gao B, Zhang H, Woo WL, Tian GY, Bai L, Yin A. Smooth nonnegative matrix factorization for defect detection using microwave nondestructive testing and evaluation. *IEEE Trans Instrum Meas* 2014;63:923-34. DOI
43. Ahmed J, Gao B, Woo WL, Zhu Y. Ensemble joint sparse low-rank matrix decomposition for thermography diagnosis system. *IEEE Trans Ind Electron* 2021;68:2648-58. DOI
44. Song J, Gao B, Woo W, Tian G. Ensemble tensor decomposition for infrared thermography cracks detection system. *Infrared Physics & Technology* 2020;105:103203. DOI
45. Ahmed J, Gao B, Woo WL. Sparse low-rank tensor decomposition for metal defect detection using thermographic imaging diagnostics. *IEEE Trans Ind Inf* 2021;17:1810-20. DOI
46. Wu T, Gao B, Woo WL. Hierarchical low-rank and sparse tensor micro defects decomposition by electromagnetic thermography imaging system. *Philos Trans A Math Phys Eng Sci* 2020;378:20190584. DOI PubMed PMC

Ayass *et al. Intell Robot* 2022;2(1):20-36
DOI: 10.20517/ir.2021.07

Intelligence & Robotics

Research Article

Open Access

Unmanned aerial vehicle with handover management fuzzy system for 5G networks: challenges and perspectives

Thalita Ayass[1], Thiago Coqueiro[2], Tássio Carvalho[2], José Jailton[2], Jasmine Araújo[1], Renato Francês[1]

[1]Institute of Technology, Federal University of Pará, Belém 66075-110, Brazil.
[2]Computer Faculty, Federal University of Pará, Castanhal 68746-360, Brazil.

Correspondence to: Dr. Thalita Ayass, Institute of Technology, Federal University of Pará, Augusto Corrêa Street, 01, Guamá, Belém City, Pará State 66075-110, Brazil. E-mail: thalita_ayass@hotmail.com

How to cite this article: Ayass T, Coqueiro T, Carvalho T, Jailton J, Araújo J, Francês R. Unmanned aerial vehicle with handover management fuzzy system for 5G networks: challenges and perspectives. *Intell Robot* 2022;2(1):20-36. https://dx.doi.org/10.20517/ir.2021.07

Received: 30 Aug 2021 **First Decision:** 22 Oct 2021 **Revised:** 8 Nov 2021 **Accepted:** 10 Feb 2022 **Published:** 22 Feb 2022

Academic Editors: Simon X. Yang, Jianjun Ni **Copy Editor:** Xi-Jun Chen **Production Editor:** Xi-Jun Chen

Abstract

The next generation of wireless networks, 5G, and beyond will bring more complexities and configuration issues to set the new wireless networks, besides requirements for important and new services. These new generations of wireless networks, to be implemented, are in extreme dependence on the adoption of artificial intelligence techniques. The integration of unmanned aerial vehicles (UAV) in wireless communication networks has opened several possibilities with increased flexibility and performance. Besides, they are considered as one of the most promising technologies to be used in the new wireless networks. Thus, UAVs are expected to be one of the most important applications to provide a new way of connectivity to the 5G network, and it is expected to grow from being a 19.3 billion USD industry in 2019 to 45.8 billion USD by 2025. In this paper, we provide a proposal of handover management on aerial 5G network utilizing the fuzzy system. The simulations performed prove the benefits of our proposal by QoS/QoE (quality of service/quality of experience) metrics.

Keywords: UAV, FANET, drone, fifth generation, fuzzy system, handover

1. INTRODUCTION

Unmanned aerial vehicles (UAVs) are considered as an interesting technology recently, mainly because of their deployment advantages and mobility[1]. There are two types of UAVs: fixed-wing and multi-rotor. The first is better applied to military applications, while the second is applied to provide wireless coverage to ground users. Moreover, UAVs are being applied in the following areas: efficient crop monitoring, delivery of goods, intelligent monitoring of places for security, carrying out surveys of various locations, developing a real-time map, coverage in telecommunications areas, and so on[2]. In addition, UAVs can support many Internet of Things (IoT) applications by providing real-time, accurate sensing/monitoring data[2].

UAVs as aerial base stations will be an essential module for future wireless technologies, as they can support high data rate transmission for users located in disaster situations (*e.g.*, after earthquakes, terrorist attacks, and so on) and when there is no typical cellular infrastructure.

Cellular networks are considered to be an alternative for drone communications because most commercial UAV systems employ IEEE 802.11 WLAN technology for sensor data, commands, and control, which operates in the unlicensed spectrum raising issues such as reliability and security[3]. Moreover, cellular networks with UAV-mounted base stations can enhance cellular networks, offering services where the traditional networks do not due to, *e.g.*, costs. Besides all the applications cited above, UAVs as aerial base stations could be promptly dispatched, cheaply maintained, and easily maneuvered. UAVs can be used as end devices through cellular networks too[4]. Thus, UAVs could benefit the current network infrastructure, in terms of coverage, reliability, and security. There are also some ongoing standardization activities (security monitoring, rescue services, *etc.*) with UAVs[4].

Some challenges to using UAVs as a main part of future mobile communications networks, serving as mobile users or mobile base stations, are interference, special mobility, and handover management. Unlike terrestrial networks, UAVs are mobile devices that move in a three-dimensional (3D) environment, which further complicates mobility issues[5] as moving to a new location would disconnect the current users. Despite these problems, UAVs are becoming important for aerial communication[6,7]. Although UAVs offer numerous benefits for future wireless communication networks, their handover is a concern that must be studied deeply[5].

The future generation wireless networks will be extremely dense and heterogeneous (with different technologies), likely equipped with moving and flying BSs (base stations). This makes the existing network planning techniques, which are mainly static and designed based on expensive field tests, not suitable for the future wireless networks. The utilization of artificial intelligence (AI) techniques for network planning has recently received interest in the research community. UAVs as aerial base stations for cellular networks are commonly used to support wireless coverage. However, an intelligent handover method must be proposed for UAV networks for when handover is triggered for a device moving to different UAVs.

One of the key premises in this development is the integration of AI into mobile communication networks. In this context, AI and machine learning techniques are expected to provide solutions for the various problems that have already been identified when UAVs are used for communication purposes such as channel modeling, resource management, positioning, interference from the terrestrial node, and handover.

This paper is structured as follows. Flying ad hoc network (FANET) concepts are discussed in Section 2. Section 3 outlines the FANET challenges and perspectives. The related works are presented in Section 4. Section 5 describes the proposed handover management by the fuzzy system in detail and the results obtained from the simulation. Section 6 summarizes the conclusion and makes suggestions for future work.

2. FLYING AD HOC NETWORK

Ad hoc networks, referred to by the IETF (Internet Engineering Task Force) as MANET (mobile ad hoc networks), have as their main characteristic the fact that they do not have infrastructure. As a result, all their functions must be performed by the devices. Thus, the devices that make up an ad hoc network must be able to communicate with each other acting as routers[1].

Ad hoc networks are often used in scenarios where there is a need to quickly set up a network, usually where there is no proper infrastructure. The devices can move arbitrarily, unpredictably modifying the network topology, which requires a permanent adaptation and reconfiguration of routes so that the devices can still communicate with each other.

In the new context of fifth-generation networks (5G), a derivation of the ad hoc networks called FANET has emerged. FANETs are ad hoc networks composed of remotely controlled flying devices (UAVs) that communicate with each other[2]. Due to the flexibility, versatility, and even easy operation of FANETs, they are used for both military and civil applications, for example plantation control in agriculture, forest clearing, and city security (see Figure 1).

In recent years, because of technological advances in areas such as robotics, telecommunications, and computer networks, UAVs have emerged as alternatives in civil and military areas, providing several applications. Thus, UAVs are intended to improve or create a network infrastructure in places that are difficult to access, such as natural disaster areas or enemy territories. With this, FANET appears as an acceptable solution in this new context, allowing the collection of information in a flexible, fast, and reliable way.

One of the goals of FANETs is to create a cooperative network, using multiple UAVs to cover an area that cannot be covered by a single UAV. Thus, it is possible to create an aerial mesh network in which its devices (drones) communicate and transmit information with each other. Therefore, it is necessary to have reliable and stable communication between devices to maintain good levels of quality of service/quality of experience (QoS/QoE).

In FANETs, the mobility index is much higher than a traditional ad hoc network, leading to frequent topology changes. This is the reason FANETs must be self-configuring and self-organizing. Such a network must be prepared for sudden changes in its topology, organization, and even communication.

The mobility of UAVs and their spatial location are also very important for determining communication routes. With the motion, these routes are usually remade to continue with the interconnection of the UAVs. For this reason, routing must be done dynamically, and the routing protocol must be efficient and simple, increasing the autonomy of the UAVs and reducing the delay in data delivery between the drones.

UAVs are responsible for overflying the environment. They have sensors to collect information and can establish communication with each other more easily by finding fewer obstacles in their line of sight, reducing the number of UAVs needed to cover a certain area. However, weather conditions can impair communication due to wind, rain, and other factors.

FANETs have a high computational power. Thus, they have a greater capacity for transmitting information since in many cases they are responsible for transmitting information in real time (with videos from the monitored environment). Thus far, there are no specific routing protocols for FANETs. Traditional

Figure 1. Flying ad hoc network scenario for city security monitoring.

protocols (AODV, OLSR, *etc.*) used in ad hoc networks are also used in this context [Figure 2].

Therefore, the location of devices within a coverage area directly impacts the performance of the network, which may improve or deteriorate according to their mobility. Thus, one of the main challenges to be solved in this type of network is the handover management after motion of UAVs.

Communication between UAVs depends a lot on their location since all information collected from the environment is concentrated in a relay node (which is responsible for relaying the data to a control center). Therefore, its positioning with respect to the other nodes is a strategic point to maintain a good performance of the network; it is not ideal that a UAV relay has excellent communication with some UAVs but poor communication with others in the network (see Figure 3).

Due to the high rate of mobility of UAVs in a FANET, updating the location of all nodes in the network is a critical factor. Network devices need to know the location of the other elements in real time; thus, in addition to the use of GPS (which on average sends the location once per second), UAVs have an inertial measurement unit, capable of sending its location in an interval smaller than the GPS at any time.

3. CHALLENGES AND PERSPECTIVES

The technology used in UAVs has great advantages for current generation telecommunications networks and provides a great framework of improvements, challenges, and promises for the next generations of wireless communications, especially in areas with difficult access or in regions lacking physical infrastructure, providing a structure and ensuring connectivity where terrestrial devices may fail. Many smart solutions are proposed in the literature, involving the context of using drones and UAVs. Some of these solutions promise implementations to adequately serve numerous services in addition to Internet data network communications and distribution, *e.g.*, surveillance services, military systems, intelligent traffic control and distribution, and other important points including in the concept of smart cities.

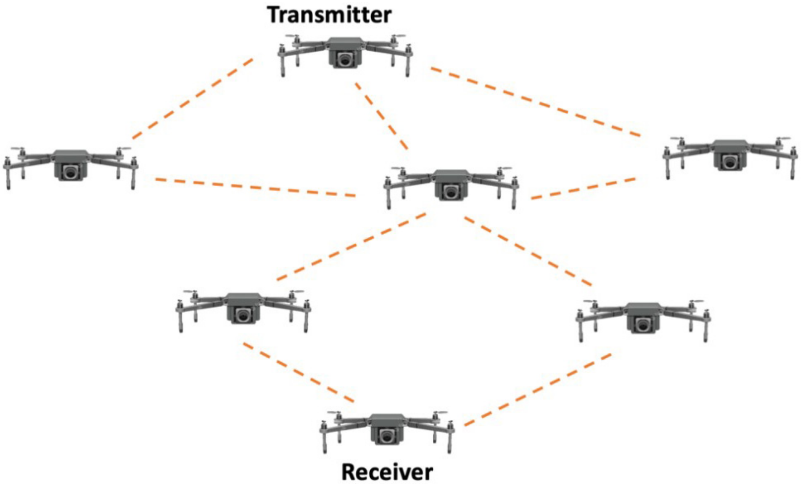


Figure 2. Flying ad hoc network routing protocol.

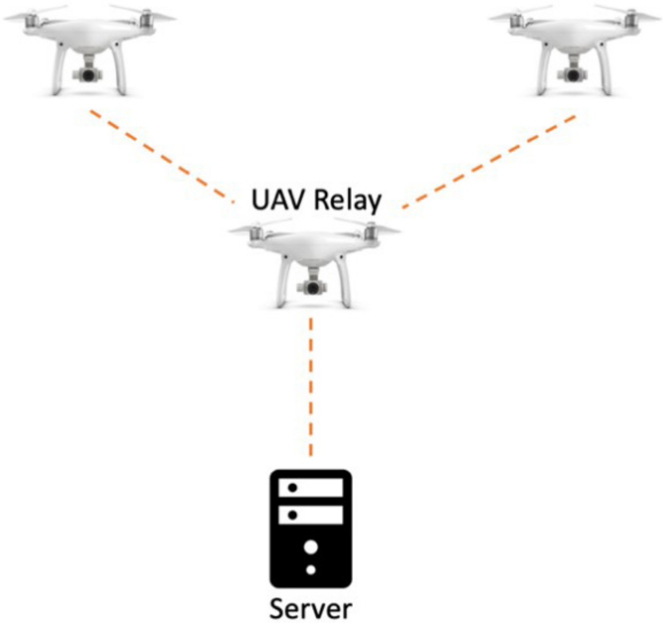


Figure 3. Unmanned aerial vehicle relay communication.

Although these systems have many advantages and benefits, they also present numerous challenges and perspectives, often due to inadequate or obsolete implementations, without updates or improvements.

3.1. Challenges

Communication networks through UAVs in the context of data communication and distribution present numerous challenges:

(1) Communication and transmission protocols are limited, due to the protocols currently used in communication networks and the Internet being obsolete for this new type of network and the way data are transmitted, with new characteristics.

(2) There is an increased probability of errors and loss of information and data due to interference and signal strength problems, impairing communication, data packet delivery, and network system reliability.

(3) There are mobility management challenges, due to the high mobility of these devices, constant change of topology, and challenging coverage management and control processes. A sub-problem in this case is related to the increase in altitude, which can generate other challenges.

(4) The reliability of communication and handovers between devices and equipment, also due to the high mobility of UAVs, is lower, which can increase the delay, impair wireless communication, make it more difficult to maintain communication links with higher quality, and bring new problems in the heterogeneity of this type of networks, especially in the context of technologies linked to 4G and 5G. UAVs experience dynamic channel swings and sudden changes due to high mobility and have constant problems with handover and ping-pong effects.

(5) Challenges regarding the battery capacity of the devices, their replacement, the transformation of the network and communications topology, and the computational and communications cost because of these constant changes, among others, also exist[8].

3.2. Perspectives

However, in another direction, communications through UAVs also have numerous proposals and possibilities for the future, as is constantly observed in current academic works, providing new possibilities for multi-hop scenarios, which allow communication services for fixed and mobile devices and the creation of new scenarios and dynamic ranges, quickly and reliably.

Among the possibilities that go beyond a communication system, we can highlight: (1) the use of UAVs for people with special needs, providing visual information, among others, for those who need it; (2) delivery services, constantly speeding up the competitive system of delivery of letters and products or assisting in this type of need; (3) environmental monitoring systems, with sensors for agriculture, water resources, temperature, and other monitoring systems, providing intelligent and dynamic decision-making; (4) offering an important resource in military scenarios or places without infrastructure, including serving as a base in scenarios of DTNs (delay tolerant networks); and (5) intelligent transport systems, helping to monitor and control traffic, accidents, and other unexpected scenarios using UAVs. Other benefits include inspection of electrical systems, use in telepresence and telemedicine, assistance in disaster and accident scenarios, smart cities, *etc*.

Thus, it can be said that the use of UAVs in wireless networks is contributing and taking network communications to a new level, integrating existing 4G and 5G networks with mobile device systems that dynamically and constantly recreate new scenarios, providing topologies, greater ranges and transmission rates, airbase station services, supporting terrestrial communications networks, helping in communication between devices and IoT environments in healthcare systems, transport with the accident detection, communication between vehicles, and energy management[9].

4. RELATED WORK

This section describes related published work on handover decision techniques on UAV networks. These are mainly about strategies to ensure an efficient handover to maintain service continuity and acceptable performance in delivering content to users.

Hu *et al.*[10] proposed an intelligent handover control method for UAVs in cellular networks. They introduced a deep learning model to predict the user's trajectory to provide mobility management. The handover decision is conceived by calculating the received signal power based on the predicted location and the characteristics of the air-ground channel, for accurate decision making. The simulation results demonstrate that the proposal's handover success rate was 8% higher than the traditional handover method.

Lee *et al.*[11] emphasized that the traditional handover decision is not suitable for drones that move and communicate in 3D space. The drone's characteristics are considered as input parameters, namely the speed limit and coverage area, which are used as input in a fuzzy system for decision making on the handover. Thus, the calculation of the number of handover decisions showed that considering the parameters related to the terminal (drone) and the parameters related to the network has a positive effect on the handover decision.

Madelkhanova *et al.*[12] developed a new algorithm based on Q-learning to manage the handover between airbase stations and static BSs, to maximize the total capacity of the UEs served by the air BSs. The Q-learning agent states are described in terms of the load of the ground bases and the reward function is defined in terms of the capacity of the UEs served by the air BSs. The results show an increase in the capacity of the UEs by up to 18% and 20% in the level of satisfaction with the solution. They also demonstrated that the Q-learning process converges quickly and only dozens of handovers are needed to achieve a significant gain.

Park *et al.*[13] presented an efficient handover mechanism for aerial networks in three-dimensional space, which differs considerably from conventional two-dimensional schemes. The proposed scheme adjusts the height of a drone and the distance between drones. For this purpose, the probability of successful handover without interruption and the false probability of starting the handover were considered to evaluate the ideal coverage decision algorithm. The proposed method was the first attempt to offer a handover scheme for drones in three-dimensional space.

Bai *et al.*[14] pointed out that the support of drones in cellular networks has allowed a wide range of new applications for next-generation wireless systems. However, they discussed that these networks were traditionally designed to serve terrestrial users, which contributes to the emergence of challenges to support wireless communication by drones. As these devices experience increased interference and channel fluctuation, they must perform handover more frequently and are more susceptible to failure rate and ping-pong during movement.

Faced with these challenges, the authors proposed an improved mobility management algorithm for drones, exploring pre-configured flight path information and their air channel properties. That is the proposal of a route-aware handover decision algorithm to minimize the failure and reduce the number of unnecessary handovers. The simulation results also demonstrate that the algorithm can reduce the ping-pong effect in certain cases.

Dong *et al.*[15] proposed a scheme that dynamically adjusts the HO trigger parameters (handover) to reduce the number of unnecessary transfers. The scheme specifically considers the UAV sailing at a certain altitude and taking off. Experiments showed that the presented solution can significantly reduce the number of unnecessary HOs and improve network performance. They also showed that the channel quality between the UAV and the BS is very different from that on the ground, and therefore selecting the most appropriate target BS is also important.

Goudarzi *et al.*[16] stated that the main challenges of communications assisted by UAVs today are to have adequate accessibility in wireless networks through mobile devices with an acceptable quality of service based on user preferences. To this end, they presented a new method based on game theory to select the best UAV during the HO process and optimize the transfer between UAVs, decreasing end-to-end delay, transfer latency, and signaling overhead. The results demonstrate the effectiveness of the proposed approach in terms of handover numbers, cost, and delay.

Azari *et al.*[17] recommended a machine learning-based approach for the HO mechanism and resource management for cellular-connected drones. They offered a machine learning-based solution that captures the correlations in temporal and spatial levels to help make HO decisions. Peng *et al.*[18] proposed a solution based on machine learning for predicting node mobility. They used the classification of movements to different classes based on predicting the nodes near a future location.

In the work of Angjo *et al.*[5], the handover decision is optimized gradually using Q-learning to provide efficient mobility and ping-pong support. The proposed scheme reduces the total number of handovers. Simulation results demonstrate that the proposed algorithm can effectively minimize the handover cost in a learning environment.

To avoid the ping-pong handover, Shakhatreh *et al.*[19] proposed a weighted fuzzy self-optimization (WFSO) approach for the optimization of the handover control parameters. The HO decision relies on three considered attributes: signal-to-interference-plus-noise ratio, the traffic load of serving and target base station, and user equipment’s velocity. The results indicate that the proposed WFSO approach significantly lowers the rates of HOPP, radio link failure, and HOF in comparison with the other algorithms found in the literature.

In this way, several studies have been conducted to address various types of HO issues, mainly in support of mobility management to reduce handover failures as well as to reduce the ping-pong effect. The ping-pong effect is the frequent connections and disconnections with the BS as the served device changes locations.

However, few proposals support energy efficiency. Battery capacity is one of the main limitations, becoming a critical factor for the continuity of the application. Therefore, effective power management is required for devices that operate on battery power. Some solutions such as wireless charging, solar charging technology, and even artificial intelligence techniques[9] are indicated for effective energy management that provides longer missions.

Finally, Singh *et al.*[20] proposed reinforcement learning (RL) based on an energy-aware ABS deployment algorithm. Dynamic movements of ABSs are managed by defining the user mobility pattern. However, this study does not support the quality of experience.

In this way, another critical factor would be the quality of user experience because it can measure the degree of quality of service through the user’s perception. It is noteworthy that expectations about the satisfaction of different services and applications vary among different users. This means that QoE is an important attribute to be considered in the handover decision-making process.

Furthermore, research work carried out in recent years has focused on the field of artificial intelligence. Approaches based on machine learning and deep learning can ensure improvements in handover decision making and save computational costs[8]. On the other hand, handover decisions consider several parameters instantly, which advantages a fuzzy approach[8,11].

Based on the survey of related literature (see Table 1), it is noted that the applicability of traditional handover schemes, as well as new propositions that support energy consumption and QoE, are still poorly investigated.

Thus, this paper proposes a fuzzy system strategy for handover decision, that is, combining support for mobility level, battery, displacement direction, throughput, and received signal strength indication (RSSI), proving that a fuzzy system is a promising technique for contributing to UAV networks.

5. FANET FUZZY SYSTEM EVALUATION

UAVs comprise a significant part of future wireless communication networks, acting as a mobile base station. While these devices provide several solutions related to mobile communication networks, UAVs also have numerous challenges, especially when it comes to handover management. Unlike terrestrial networks, drones are mobile devices that move in a 3D environment, which further complicates mobility issues.

Handover is one of the essential processes in wireless communication networks that guarantee continuous connection and quality of service while users are mobile. The criterion for the conventional handover decision is based primarily on the RSSI to indicate whether the device will remain attached to the current point of access or not. In the context of FANETs, this single premise for network selection can result in failures or even interruptions in service, since the UE can connect to a UAV with a low battery level, which is one of the most critical factors in these devices.

Similarly, high user mobility can compromise the quality of experience, due to the excessive number of handovers and the “ping-pong” effect that can direct the UE to a saturated network that offers low bandwidth.

Given this context, this work contributes with a study case that consists of presenting a system based on fuzzy logic as it is widely used in dynamic scenarios, as in the case of networks composed of UAVs, to assist in the decision making of handover in a FANET. The fuzzy system considers three input parameters: user speed, RSSI, and drones battery level. These inputs are processed by the inference system for the defuzzifier to evaluate and generate the decision-making outputs.

In fuzzy systems, the results are classified into a range from 0 to 1. A value of 0 denotes an absolute exclusion, while a value of 1 denotes a complete correlation. The gap between the two extremes results in intermediate degrees of relevance. Elements can also belong to two or more defined sets, observing the values of the membership functions for each element.

One or more linguistic variables can be associated with the set of Fuzzy values, which represent the universe of the possibility of the expected results. In this work, the terms used to classify the outputs with the possibility of triggering the handover are: no, probably no, probably yes, and yes. The handover process is executed when the inference value is equal to or greater than 0.6.

The system considers three input parameters that are processed by the inference system so that the defuzzifier can evaluate and generate the decision-making outputs. The first is related to the user’s level of

Table 1. Related works

Paper	Proposed solution	Ping-pong handover reduction	Energy efficiency support	Mobility management support	QoE support
[10]	*Deep learning*	No	No	Yes	No
[11]	Fuzzy	No	No	Yes	No
[12]	Q-*learning algorithm*	No	No	Yes	No
[13]	Coverage decision algorithm which controls the coverage of each net-drone	No	No	Yes	No
[14]	Route-aware handover algorithm	Yes	No	Yes	No
[15]	Dynamic parameters to handover decision	No	No	Yes	No
[16]	Cooperative game theory	Yes	No	Yes	No
[17]	Machine learning-based solution	No	No	Yes	No
[18]	Machine learning-based solution	No	No	Yes	No
[5]	Q-learning based	Yes	No	Yes	No
[19]	Fuzzy system	Yes	No	Yes	No
[20]	Reinforcement learning	No	Yes	Yes	No

QoE: Quality of experience.

mobility and indicates how long a mobile device remains in the coverage area of a station. The faster the device travels, the less time it will be connected to that access point. This first input is divided into three sets of linguistic values: slow (range 0-1.5 m/s), moderate (1.3-3 m/s), and fast (2.5-4 m/s).

The second input refers to the received signal level, represented by RSSI. This is a factor used to assess how likely the device is to disconnect from the access point if the signal strength is weak. In this metric, signal levels are defined for language sets as follows: weak (-120 to -100 dBi), moderate (-115 to -65 dBi), and strong (> -72 dBi).

The last input metric considers the drone's flight range, which is linked to how long the devices can remain in operation. This is an important criterion because, given the knowledge of the remaining time each UAV can still operate, unnecessary transfers are avoided for those drones that are in the unloading phase and will not be able to continue the service. For this parameter, the defined sets are: low (0-10 min), medium (8-20 min), and high (18-30 min) battery levels.

Given the inputs, the fuzzy inference system will determine the outputs according to the set of 27 rules previously established from the combination of the three parameters. In this work, the Gaussian membership function is applied to all inputs and outputs. This function is chosen because of its characteristic of reducing the noise of input variables and its ability to represent real-world phenomena more naturally.

The output of the fuzzy system indicates the probability of the mobile device starting the handover process. In general, if a user has high mobility and high levels of RSSI, the transfer process to another network will not occur. The system indicates a trend of execution of the handover, as its inference value is equal to 0.6.

In the 3D surface graphics in Figure 4, it is possible to visualize the relationship between the chosen parameters. The region in blue corresponds to a user with high mobility and excellent signal strength. In this context, the handover process will not trigger. The yellow region indicates the opposite, the user with low speed and receiving a bad signal; in this case, the handover is executed.

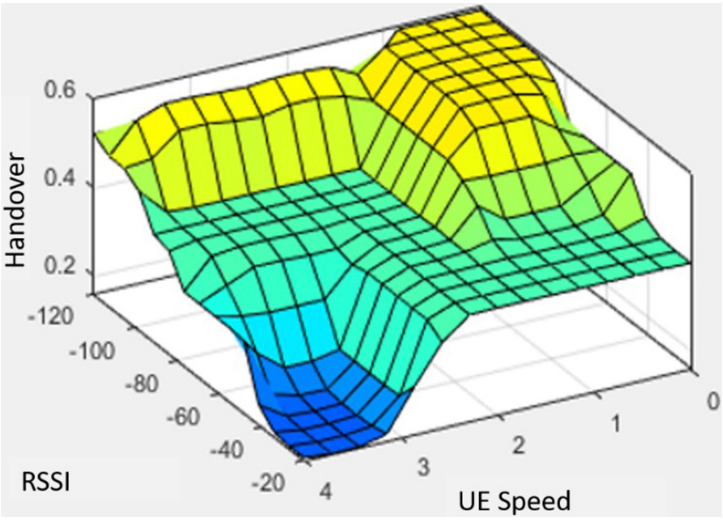


Figure 4. Inference fuzzy system.

The fuzzy system was implemented using the Matlab Fuzzy logic toolbox, where its inputs were defined and, after going through the defuzzification process, produced the numerical outputs that indicated the tendency to carry out the handover or not. To evaluate the performance of the network, according to the outputs that were indicated by the fuzzy system as being ideal for the handover decision, the technique used was to implement the scenarios in the simulation environment of the Network Simulator 2 (NS2) tool. The UAVs were placed at the same height of 100 m, in an area of 1000 m × 1000 m, as shown in Figure 5. In the simulation, a WI-FI network is considered where the UAVs serve as access points to promote the connection of users within a given environment, according to the displacement of the UEs. The main parameters used in simulation are summarized in Table 2.

To better understand the results, the evaluation considered the network throughput metric to verify the behavior of the proposal through the solution that was based on fuzzy logic for handover decision making.

In a first scenario, CBR-type applications were received by mobile users through the WI-FI interface enabled by UAVs that are operating as a network access point. The scenario was simulated by comparing the traditional handover process, which prioritizes RSSI as a transfer trigger, and handover from the proposed fuzzy architecture.

It was considered a high mobility environment within the UAVs' coverage area. In this context, it can be seen from the graph in Figure 6 that, by the traditional handover method, the UEs were subject to the ping-pong handover effect and suffered a lot of instability in the connection. This behavior is perceived by the fact that the conventional handover model does not consider parameters that are characteristic of FANETs, especially regarding the UAV battery.

It is noticed that, between Seconds 90 and 120 of the simulation, there was an interruption in the service, caused by the unloading of the UAV. Even though the UE is reconnected from Second 120, right after the device suffers another disconnection because, even with good signal strength, the UAV was in full unloading phase. Differently, the handover proposed using the fuzzy system parameters that meet the characteristic requirements of UAVs, such as battery time, proved to be efficient when selecting a new network.

Table 2. Simulation parameters

Parameter	Value
UAV	12
Access technology	IEEE 802.11 g
Propagation model	Shadowing
Pathloss	2 (dB)
Shadowing deviation	4 (dB)
Mobility model	Random waypoint
Application	CBR/Video
Rate	54 Mbps
Simulation time	130 s

UAV: Unmanned aerial vehicle.

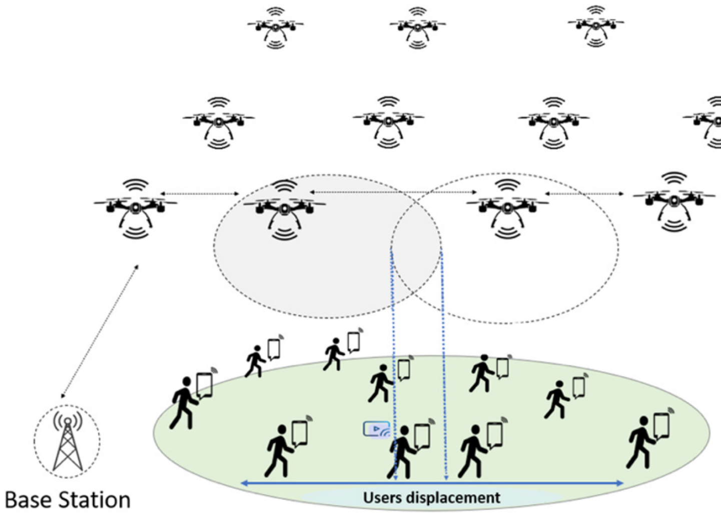


Figure 5. Flying ad hoc network scenario simulation.

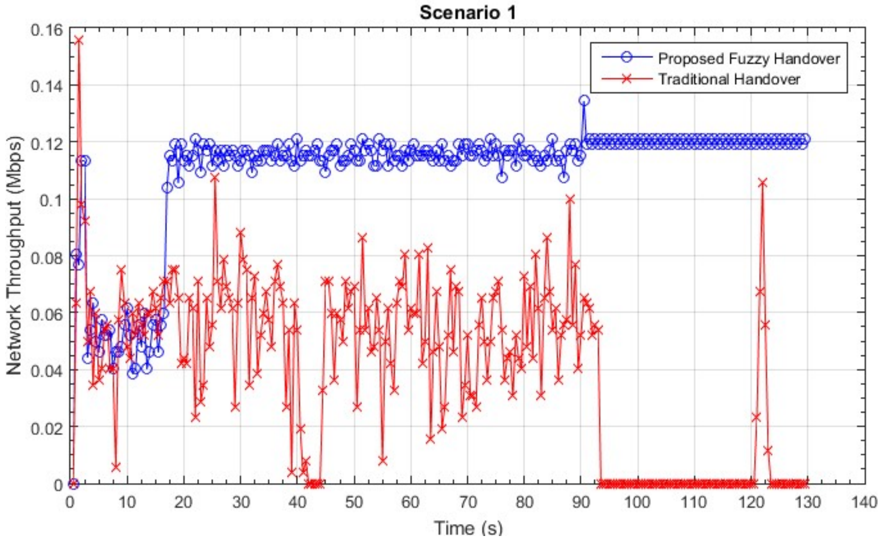


Figure 6. Throughput of Network I.

From the criteria established in the architecture, it is noted that at around Second 19 of simulation the handover is performed, causing the average flow rate to reach twice the value of the previous connection and remain stable until the end of the simulation. This better performance is possible due to the management made by the proposal of handover with the fuzzy system, which prevents the UE from selecting a new access point where the UAV is about to discharge, even if it presents good signal strength.

In the second scenario illustrated in Figure 7, the network was subjected to a greater demand for data due to the increase in the number of users overloading the network. It is possible to see that, without the proposed solution, the network presented an even worse performance than in the previous scenario, where the flow rate drops drastically when using the traditional handover model.

The running application is also CBR type, and, by the traditional model, the transfer was made to the nearest network, even though there was no interruption in the connection. The new network was more overloaded and ended up causing the throughput to be below 0.1 Mbps. Conversely, the proposed method performed the handover only when necessary and maintained a stable connection when selecting a better network.

As in the first scenario, the handover with intelligent management of parameters by the fuzzy system was more efficient as it managed to maintain a constant connection, in addition to identifying the best access point and promoting a better flow rate to the UE from Second 90 of the simulation.

The study also analyzed the effects of the traditional process of handover and the one proposed by fuzzy inference, through simulations involving video application. To evaluate the quality of the media received, the QoE results in the same previous scenarios were compared. The video used in the simulation has a resolution of 176 × 144, 1000 frames, and decoding in YUV 4:2:0 format, which stands for the color difference encoding system whether composite or component.

In this way, the peak signal-to-noise ratio (PSNR), structural similarity metric (SSIM), and video quality metric (VQM) metrics were considered, being these objective metrics that complement each other, to assess the impacts of signal degradation in the original video with the reference when the traditional handover was performed, as well as the proposed one. At the end of the transmission, the values of the metrics in question were calculated and displayed frame by frame.

PSNR has a range of values between 0 and 100 dB. For values above 30 dB, it is understood that the video has good quality. On the other hand, videos that are below the 20 dB range are considered of poor quality. For the network in the first scenario, Figure 8 shows the PSNR values for each frame of the video. Comparison with the original file shows that the PSNR of the video received had an average of 21.3 dB in the traditional handover, classifying it as a low-quality video. Differently, the PSNR for the video with the Fuzzy criteria obtained an average of 42.41 dB, characterizing it as being of excellent quality.

The range of SSIM values is between 0 and 1, where 1 represents the exact correlation with the original image. Figure 9 shows the SSIM values of frames in the traditional handover obtaining an average of 0.59. For the proposed handover, the result obtained was 0.98, being very close to 1, which is the accepted value for a perfect correlation of images. In this sense, it could be seen that the reference video had low distortion for this parameter.

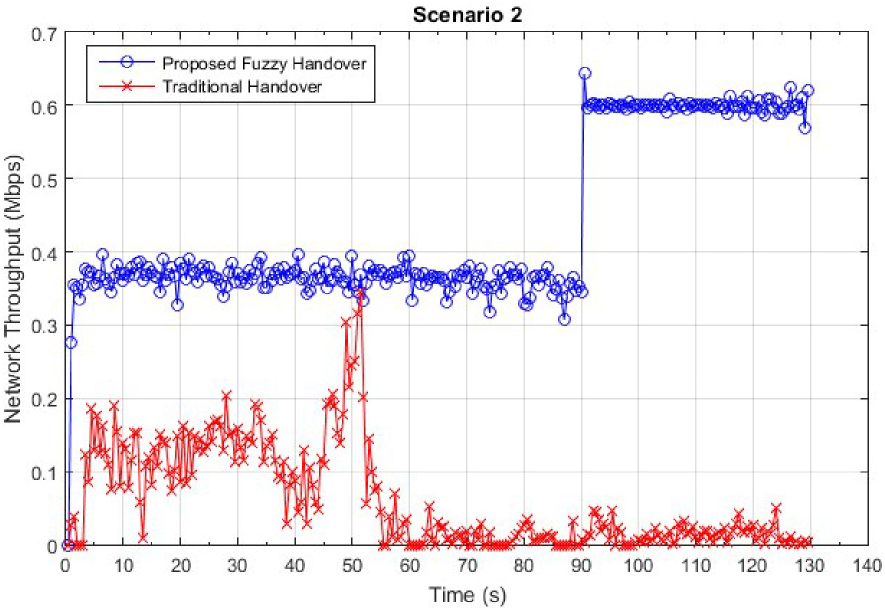


Figure 7. Throughput of Network II.

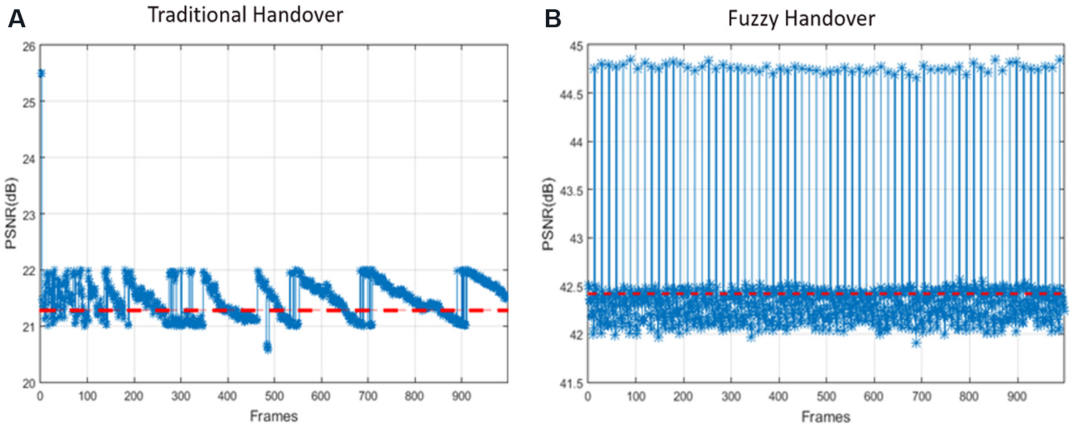


Figure 8. Peak signal-to-noise ratio traditional (A) and fuzzy handover (B).

The evaluation of the video quality of VQM is illustrated in Figure 10. For this metric, 0 characterizes the best possible value. The result obtained after the evaluation shows that the average was equal to 6.01 for the handover without the fuzzy system. The proposal achieved an average VQM of 0.50, which indicates that the video did not suffer considerable degradation in this sense, being close to the ideal value of 0.

The superiority in the quality of the transmitted video that considers the HO by the fuzzy system can also be seen visually, making a frame-by-frame comparison between the original and received video in the two methods covered in the work. In Figure 11, Frame 305 presents degradations and distortions in the pixels of the video transmitted over the network without supporting the efficient handover scheme. However, the same frames remained undistorted in the video that was broadcast considering the fuzzy proposal.

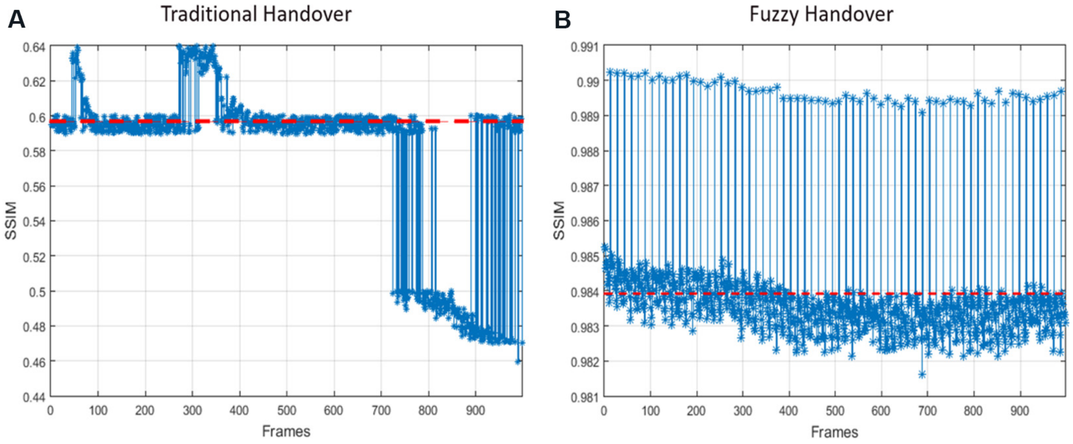


Figure 9. Structural similarity metric traditional (A) and fuzzy handover (B).

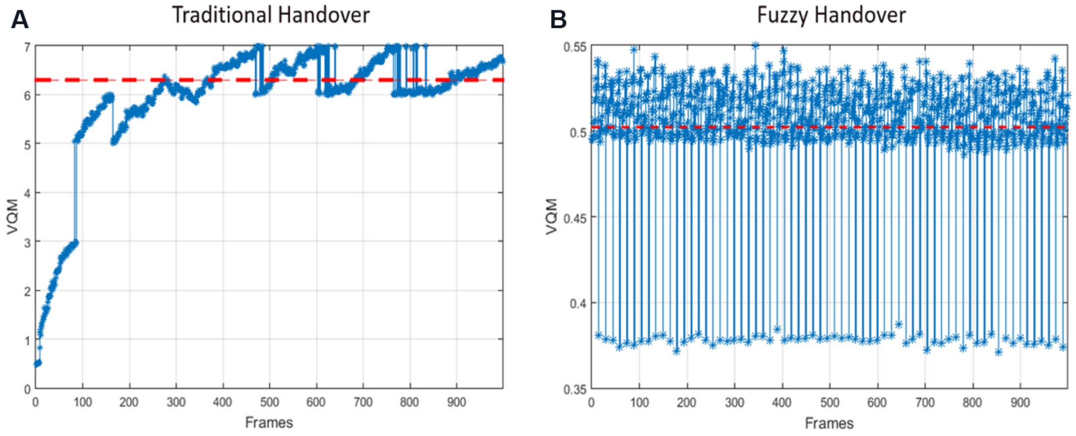


Figure 10. Video quality metric traditional (A) and fuzzy handover (B).

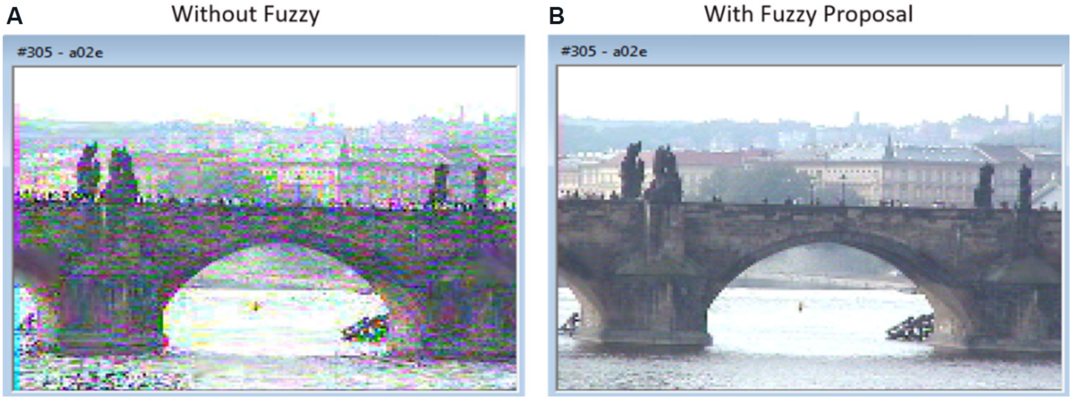


Figure 11. Frames comparison without fuzzy (A) and with the fuzzy proposal (B).

From the simulations and evaluative analysis on the performance of both the conventional handover model and the proposed one, it is possible to see the traditional model does not satisfy the peculiarities of these networks and their components, causing the low quality of service, compromising the quality of experience of the user, and not guaranteeing a transparent handover, which impacts the continuity of the service.

By the proposed architecture, the handover process proved to be effective to mitigate handover issues in FANETs, since it achieved superior results in both QoS and QoE parameters, proving itself as a viable proposal.

6. CONCLUSIONS

In the context of fifth-generation networks, new connectivity alternatives emerge, such as the use of UAVs as an access point, mainly for locations with difficult access or without available network infrastructure.

To set up fast and temporary networks, FANETs are used in different scenarios to provide 5G access to users, but, due to the mobility of UAVs and the users themselves, the network topology is constantly modified. The constant topology changes in FANET due to mobility can disrupt the user's connection.

Thus, this paper proposes a FANET as an alternative way of connecting to 5G networks, in which drones work as access points for users. The paper also proposes the use of a fuzzy system for UAV mobility management for anticipating handovers to avoid network connection breaks.

The results prove a better performance when compared to a traditional FANET (without the use of the fuzzy system). The results were proven through the throughput metric and QoE metrics (for video transmissions), as well as the shown frames of transmitted videos. As future work, we intend to use other artificial intelligence techniques, as well as other wireless transmission technologies.

DECLARATIONS

Authors' contributions

Literature review, research/methodology design, writing the manuscript from initial writing until submission, revision of the manuscript according to reviewers' comments, data collection, result analysis and interpretation: Ayass T

Literature review, writing the manuscript from initial writing until submission, revision of the manuscript according to reviewers' comments, result analysis and interpretation: Coqueiro T

Formulation of the initial research proposal, research/methodology design, writing the manuscript from initial writing until submission, revision of the manuscript according to reviewers' comments, result analysis and interpretation: Carvalho T

Formulation of the initial research proposal, research/methodology design, technical guidance and assistance to methodology, writing the manuscript from initial writing until submission, revision of the manuscript according to reviewers' comments, co-researcher of the student: Jailton J

Formulation of the initial research proposal, literature review, technical guidance and assistance to methodology, writing the manuscript from initial writing until submission, co-researcher of the student: Araújo J

Formulation of the initial research proposal, technical guidance and assistance to methodology, researcher/supervisor of the project and student: Francês R

Availability of data and materials

Not applicable.

Financial support and sponsorship

None.

Conflicts of interest

All authors declared that there are no conflicts of interest.

Ethical approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

Copyright

REFERENCES

1. Carvalho T, Jailton Júnior J, Francês R. A new cross-layer routing with energy awareness in hybrid mobile ad hoc networks: a fuzzy-based mechanism. *Simul Model Pract Theory* 2016;63:1-22. DOI
2. Noorwali A, Awais Javed M, Zubair Khan M. Efficient UAV communications: recent trends and challenges. *Comput Mater Contin* 2021;67:463-76. DOI
3. Fakhreddine A, Bettstetter C, Hayat S, Muzaffar R, Emini D. Handover challenges for cellular-connected drones. DroNet'19: Proceedings of the 5th Workshop on Micro Aerial Vehicle Networks, Systems, and Applications; New York: Association for Computing Machinery. 2019. p. 9-14. DOI
4. Enhanced LTE support for aerial vehicles. Available from: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3231 [Last accessed on 17 Feb 2022].
5. Angjo J, Shayea I, Ergen M, Mohamad H, Alhammadi A, Daradkeh YI. Handover management of drones in future mobile networks: 6G technologies. *IEEE Access* 2021;9:12803-23. DOI
6. Zeng Y, Guvenc I, Zhang R, Geraci G, Matolak DW, UAV communications for 5G and beyond. Hoboken: Wiley-IEEE Press; 2020. DOI
7. Saad W, Bennis M, Mozaffari M, Lin X. Wireless communications and networking for unmanned aerial vehicles. Cambridge: Cambridge University Press; 2020. DOI
8. Li B, Fei Z, Zhang Y, Guizani M. Secure UAV communication networks over 5G. *IEEE Wireless Commun* 2019;26:114-20. DOI
9. Sharma A, Vanjani P, Paliwal N, et al. Communication and networking technologies for UAVs: a survey. *J Netw Comput Appl* 2020;168:102739. DOI
10. Hu B, Yang H, Wang L, Chen S. A trajectory prediction based intelligent handover control method in UAV cellular networks. *China Communications* 2019;16:1-14. DOI
11. Lee E, Choi C, Kim P. Intelligent handover scheme for drone using fuzzy inference systems. *IEEE Access* 2017;5:13712-9. DOI
12. Madelkhanova A, Becvar Z. Optimization of cell individual offset for handover of flying base station. 2021 IEEE 93rd Vehicular Technology Conference (VTC2021-Spring); 2021 Apr 25-28; Helsinki, Finland. IEEE; 2021. p. 1-7. DOI
13. Park K, Kang J, Cho B, Park K, Kim H. Handover management of net-drones for future internet platforms. *Int J Distrib Sens Netw* 2016;12:5760245. DOI
14. Bai J, Yeh SP, Xue F, Talwar S. Route-aware handover enhancement for drones in cellular networks. 2019 IEEE Global Communications Conference (GLOBECOM); 2019 Dec 9-13; Waikoloa, HI, USA. IEEE; 2019. p. 1-6. DOI
15. Dong W, Mao X, Hou R, Lv X; Li H. An enhanced handover scheme for cellular-connected UAVs. 2020 IEEE/CIC International Conference on Communications in China (ICCC); 2020 Aug 9-11; Chongqing, China. IEEE; 2020. p. 418-23. DOI
16. Goudarzi S, Anisi MH, Ciuonzo D, Soleymani SA, Pescape A. Employing unmanned aerial vehicles for improving handoff using cooperative game theory. *IEEE Trans Aerosp Electron Syst* 2021;57:776-94. DOI
17. Azari A, Ghavimi F, Ozger M, Jantti R, Cavdar C. Machine learning assisted handover and resource management for cellular connected drones. Available from: http://arxiv.org/abs/2001.07937 [Last accessed on 17 Feb 2022].
18. Peng H, Razi A, Afghah F, Ashdown J. A unified framework for joint mobility prediction and object profiling of drones in UAV networks. *J Commun Netw* 2018;20:434-42. DOI
19. Shakhatreh H, Sawalmeh AH, Al-fuqaha A, et al. Unmanned aerial vehicles (UAVs): a survey on civil applications and key research challenges. *IEEE Access* 2019;7:48572-634. DOI
20. Singh S, Sandhu MK. A review over existing handover decision systems for drones in wireless network. Available from: http://www.ijstr.org/final-print/mar2020/A-Review-Over-Existing-Handover-Decision-Systems-For-Drones-In-Wireless-Network.pdf [Last accessed on 17 Feb 2022].

Harib *et al. Intell Robot* 2022;2:37-71
DOI: 10.20517/ir.2021.19

Intelligence & Robotics

Review

Open Access

Evolution of adaptive learning for nonlinear dynamic systems: a systematic survey

Mouhcine Harib[1], Hicham Chaoui[1], Suruz Miah[2]

[1]Department of Electronics, Carleton University, Ottawa, ON K1S 5B6, Canada.
[2]Electrical and Computer Engineering, Bradley University, Peoria, IL 61625, USA.

Correspondence to: Prof. Hicham Chaoui, Department of Electronics, Carleton University, 7066 Minto Building, Ottawa, ON K1S 5B6, Canada. E-mail: Hicham.Chaoui@carleton.ca

How to cite this article: Harib M, Chaoui H, Miah S. Evolution of adaptive learning for nonlinear dynamic systems: a systematic survey. *Intell Robot* 2022;2:37-71. https://dx.doi.org/10.20517/ir.2021.19

Received: 19 Dec 2021 **First Decision:** 20 Jan 2022 **Revised:** 3 Feb 2022 **Accepted:** 24 Feb 2022 **Published:** 16 Mar 2022

Academic Editor: Simon X. Yang **Copy Editor:** Xi-Jun Chen **Production Editor:** Xi-Jun Chen

Abstract

The extreme nonlinearity of robotic systems renders the control design step harder. The consideration of adaptive control in robotic manipulation started in the 1970s. However, in the presence of bounded disturbances, the limitations of adaptive control rise considerably, which led researchers to exploit some "algorithm modifications". Unfortunately, these modifications often require a priori knowledge of bounds on the parameters and the perturbations and noise. In the 1990s, the field of Artificial Neural Networks was hugely investigated in general, and for control of dynamical systems in particular. Several types of Neural Networks (NNs) appear to be promising candidates for control system applications. In robotics, it all boils down to making the actuator perform the desired action. While purely control-based robots use the system model to define their input-output relations, Artificial Intelligence (AI)-based robots may or may not use the system model and rather manipulate the robot based on the experience they have with the system while training or possibly enhance it in real-time as well. In this paper, after discussing the drawbacks of adaptive control with bounded disturbances and the proposed modifications to overcome these limitations, we focus on presenting the work that implemented AI in nonlinear dynamical systems and particularly in robotics. We cite some work that targeted the inverted pendulum control problem using NNs. Finally, we emphasize the previous research concerning RL and Deep RL-based control problems and their implementation in robotics manipulation, while highlighting some of their major drawbacks in the field.

Keywords: Adaptive control, deep reinforcement learning, manipulators, neural networks, reinforcement learning, robotics

1. INTRODUCTION

By running a numerical model of a robotic mechanism and its interactions with surroundings, one can define a control algorithm that delivers torque (input) signals to actuators, and that is how a mechanism is able to anticipate the movement. Since robotic systems are extremely nonlinear, the control design is usually a hard step. Figure 1 illustrates a simplified representation of a two-link robot manipulator. Given a system of (dynamic) equation of a robotic system, it contains variables that change when the robot is in motion, which alters the equation mid-task. In this case, a traditional control technique will have to divide the nonlinear mechanism into linear subsystems, which are reasonable for low-speed actions; however, with a high-speed system, their efficacy becomes close to none. For these reasons, adaptive control strategies were first considered.

The system defined by a robot and its controller is complete. Since reconfigurations of the robotic mechanism are needed due to the functional requirements changes, the controller has to adapt to these reconfigurations. In comparison to a non-adaptive control, the adaptive control is able to function without relying on the prior data from the system, since it constantly changes and adjusts to the altered states. That is specifically what makes adaptive control “almost perfect” for systems with unpredictable surroundings, with many probable interferences that could change the system parameters anytime.

In the early years, there were many interests in research and books about the adaptive control[1-5] that considered continuous-time systems in most cases. Since 1970, researchers have started dealing with the realization of adaptive control in digital systems. Multiple surveys[6-8] show that the consideration of adaptive control systems with discrete-time signals has been around for a while. Many applications of the general adaptive control have been made afterward. There are two fundamental approaches within the adaptive control theory. The first approach is called Learning Model Adaptive Control, where we find the well-known self-tuning adaptive control technique. This approach consists of an improved model of the plant obtained by on-line parameter estimation techniques, and then used in the feedback control. The second approach is called Model Reference Adaptive Control (MRAC). In this case, the controller is adjusted so that the behaviors of the closed-loop system and the preselected model match according to some criterion[9].

Due to the limitations of adaptive control when it comes to bounded disturbances, many researchers turned to “Algorithm Modification” approaches in the 1980s. Typically, these approaches alter least squares adaptation by putting bounds on the error, the parameters, or employing a first order modification of the least squares type of adaptation algorithm. When the observed error is not attributable to an error in the parameter estimations, these strategies effectively turn off or limit the effects of parameter adaptation. The Algorithm Modification techniques essentially perform the same function as the input-output rule-based approaches, but they attempt to have the adaptation algorithm monitor its own level of certainty. The second section of this paper will present more details about the most famous modifications among control researchers, such as Dead-zone modification, σ-modification, and ϵ-modification. Unfortunately, these modifications often require a priori knowledge of bounds on the parameters and the perturbations and noise[10]. Furthermore, they often improve robustness at the expense of performance.

In a control engineering sense, AI and classical control-based approaches are just different sides of the same coin. Therefore, the limitation of Adaptive control has driven many researchers to consider AI-based controllers. In the 1990s, the field of neural networks was vastly investigated in general, and for control of dynamical systems in particular. The control problem can be formulated as a machine learning (ML) problem, and that is how ML can be mixed with control theory. One of the fundamentally new approaches

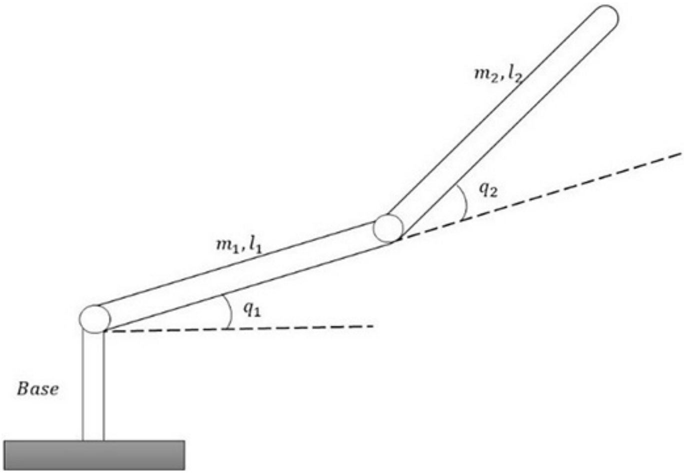


Figure 1. Simple schematic of a two-link robot manipulator.

is the PILCO approach[11].

Artificial Neural Networks (ANNs) have been explored for a long time in the hopes of obtaining human-like performance in speech and image processing. Several types of Neural Networks (NNs) appear to be promising candidates for control system applications. Multilayer NNs (MLNs), recurrent NNs (RNNs), and the cerebellar model articulation controller (CMAC) are examples of these. The decision of the NN to employ and which training technique to utilize is crucial, and it changes according to the application. The type of NNs most commonly used in control systems is the feedforward MLNs, where no information is fed back during operation. There is, however, feedback information available during training. Typically, supervised learning methods, where the neural network is trained to learn input-output patterns presented to it, are used. Most often, versions of the backpropagation (BP) algorithm are used to adjust the NN's weights during training. The feedforward MLNs are the most often employed NNs in control systems since no information is fed back during operation. During training, however, there is feedback information accessible. In most cases, supervised learning methods are utilized, in which the NN is taught to learn input-output patterns that are provided to it. During training, variants of the BP algorithm are frequently employed to change the NN weights. More details about NNs, for dynamical systems in general and for robotics in particular, are discussed in section 3 of this work.

In robotics, it all boils down to making the actuator perform the desired action. The basics of control systems tell us that the transfer function decides the relationship between the output and the input given the system or plant. While purely control-based robots use the system model to define their input-output relations, AI-based robots may or may not use the system model and rather manipulate the robot based on the experience they have with the system while training or possibly enhance it in real-time as well.

Reinforcement learning (RL) is a type of experience-based learning that may be used in robotics when on-line learning without knowledge of the environment is necessary. The controller may learn which of its possible actions will result in the greatest performance for a particular job for each of its states. If the mobile robot collides with an obstacle, for example, it will learn that this is a poor action, but if it achieves the objective, it will learn that this is a positive activity. Reinforcement or reward is the term for such contextual feedback. The goal of the controller is to maximize its predicted future rewards for state-action pairings, which are represented by action values. Q-learning is a popular type of RL in which the best policy is learned implicitly as a Q-function. There have been several publications on the use of RL in robotic systems.

This review is organized into 3 sections besides the present introductory chapter and a concluding section. In section 2, we talk about adaptive control limitations for nonlinear systems and introduce the probable drawbacks in the presence of disturbances. We also present the main modifications proposed in the 1980s to overcome these limitations. Section 3 will focus on presenting the work that implemented NNs in nonlinear dynamical systems and particularly in robotics, while we cite some work that targeted the inverted pendulum control problem using NNs. Finally, section 4 emphasizes the previous research concerning RL and Deep RL (DRL) based control problems and their implementation in robotics manipulation, while highlighting some of their major drawbacks in the field.

2. ADAPTIVE CONTROL LIMITATIONS - BOUNDED DISTURBANCES

Given a system/plant with an uncertainty set, it is clear that the control objective will be intuitively achievable through either identification, robust control, or a combination of both as in adaptation. The identification is the capability to acquire information in reducing uncertainty. This problem characterization had seen some rigorous analysis over a long period of time and can become very challenging. In 2001, Wang and Zhang[12] explored some fundamental limitations of robust and adaptive control by employing a basic first-order linear time-varying system as a vehicle. One can notice that robust control cannot deal with much uncertainty, while the use of adaptive control shows a much better capability of dealing with uncertain parameters and providing better robustness. However, adaptive control requires additional information on parameter evolution and is fundamentally limited to slowly time-varying systems. Furthermore, adaptation is not capable of achieving proximity to the nominal performance when under near-zero variation rates.

2.1. Problem statement

The design of adaptive control laws is always under the assumption that system dynamics are exactly specified by models. Hence, when the true plant dynamics is not perfectly described by any model, as expected from a practice point of view, one can only question the real behavior of the control. The robust stability, required for any adaptive control to achieve practical applicability of the algorithms, can be provided when only the modeling error is sufficiently "small". Unfortunately, stability alone cannot guarantee robustness, since the modeling error appears as a disturbance and usually causes divergence of the adaptive process.

While one of the fundamental fields of application of adaptive control is in systems with unknown time-varying parameters, the algorithms have been proved robust, in the presence of noise and bounded disturbances, only for systems with constant parameters[13]. Ideally, when there are no disturbances or noise and when parameters are constant, adaptation shows smooth convergence and stability properties. On the other hand, the adaptive laws are not robust in the presence of bounded disturbances, noise and time-varying parameters.

In order to mathematically state the problem of non-robustness of adaptive control to bounded disturbances, let us start by considering a MIMO system in the form[14,15],

$$\begin{gathered}\dot{x} = A_{ref}x + B\Lambda\left(u + K^T\Psi(x)\right) + B_{ref}y_{cmd} + \xi(t) \\ y = C_{ref}x\end{gathered} \tag{1}$$

where $\xi(t) \in \mathrm{R}^n$ is a bounded time-dependant disturbance, $x \in \mathrm{R}^n$ is the extended system state vector, $y \in \mathrm{R}^n$ is the controlled system output, $u \in \mathrm{R}^m$ is the control input and $\Psi \in \mathrm{R}^N$ is the known N-dimensional regressor vector. We assume $(A_{ref}, B, B_{ref}, C_{ref})$ are known and A_{ref} is Hurwitz. $y_{cmd} \in \mathrm{R}^m$ in this case is a bounded command for y. $\Lambda \in \mathrm{R}^{m\times m}$ is a diagonal positive definite matrix and $\mathrm{K} \in \mathrm{R}^{N\times m}$ is a constant matrix, where both matrices represent the matched uncertainties of the system. In addition, we assume that,

$$\|\xi(t)\| \leq \xi_{max} \tag{2}$$

and that the disturbance upper bound $\xi_{max} \geq 0$ is known and constant.

The control goal is bounded tracking of the reference model dynamics,

$$\begin{gathered} \dot{x} = A_{ref}x_{ref} + B_{ref}y_{cmd} \\ y = C_{ref}x_{ref} \end{gathered} \tag{3}$$

driven by a bounded time-dependant command $y_{cmd} \in \mathrm{R}^m$.

Based on Equation (1), the control input is selected as,

$$u = -\hat{K}^T \Psi(x) \tag{4}$$

where $K \in \mathrm{R}^{N\times m}$ is the matrix of adaptive parameters. If we substitute Equation (4) into Equation (1), we get,

$$\dot{x} = A_{ref}x - B\Lambda\Delta K^T \Psi(x) + B_{ref}y_{cmd} + \xi(t) \tag{5}$$

where,

$$\Delta K = \hat{K} - K \tag{6}$$

is the matrix of estimation errors. The tracking error is $e = x - x_{ref}$. Subtracting the reference model dynamics in Equation (3) from that of Equation (1) yields the tracking error dynamics,

$$\dot{e} = A_{ref}e - B\Lambda\Delta K^T \Psi(x) + \xi(t) \tag{7}$$

The Lyapunov function candidate is selected,

$$V(e,\Delta K) = e^T Pe + trace(\Delta K^T \Gamma_K^{-1} \Delta K\Lambda) \tag{8}$$

where $\Gamma_K = \Gamma_K^T > 0$ represents constant rates of adaptation, and $P = P^T > 0$ is the unique symmetric positive definite solution of the algebraic Lyapunov equation,

$$A_{ref}^T P + PA_{ref} = -Q \tag{9}$$

with $Q = Q^T > 0$. The time derivative of V, along the trajectories of Equation (7),

$$\dot{V} = -e^T Qe - 2e^T PB\Lambda\Delta K^T \Psi(x) + 2trace\left(\Delta K^T \Gamma_K^{1} \hat{K}\Lambda\right) + 2e^T P\xi(t) \tag{10}$$

Applying the trace identity,

$$a^T b = trace(ba^T) \tag{11}$$

yields,

$$\dot{V} = -e^T Qe + 2trace\left(\Delta K^T \left\{\Gamma_K^{1} \hat{K} - \Psi e^T PB\right\}\Lambda\right) + 2e^T P\xi(t) \tag{12}$$

Using the following adaptive law yields,

$$\dot{\hat{K}} = \Gamma_K \Psi e^T PB \tag{13}$$

then,

$$\dot{V} = -e^T Qe + 2e^T P\xi(t) \le -\lambda_{min}(Q)\|e\|^2 + 2\|e\|\xi_{max}\lambda_{max}(P) \tag{14}$$

and, consequently, $V < 0$ outside of the set,

$$E_0 = \left\{(e,\Delta K) \mid \|e\| \le \frac{2\xi_{max}\lambda_{max}(P)}{\lambda_{min}(Q)} = e_0\right\} \tag{15}$$

Hence, trajectories $[e(t),\Delta K(t)]$, of the error dynamics in Equation (7) coupled with the adaptive law in Equation (13), enter the set E_0 in finite time and stay there for all future times. However, the set E_0 is not compact in the $(e,\Delta K)$ space. Moreover, it is unbounded since ΔK is not restricted. Inside the set E_0, V can become positive and consequently, the parameter errors can grow unbounded, even though the tracking error norm remains less then e_0 at all times. This phenomenon is caused by the disturbance term $\xi(t)$. It shows that the adaptive law in Equation (13) is not robust to bounded disturbances, no matter how small the latter is.

In the 1980s, several studies analyzed the stability of adaptive control systems and many of them concentrated on linear disturbance-free systems[16-20]. The results, however, are not completely satisfactory, since they do not consider the cases where disturbances are present, which could completely change the efficiency of the control system, even when very small, leading to instability. In the years that followed, there have been many attempts to overcome the limitations of adaptive control in the presence of bounded disturbances. In these published papers[21-25], it is shown that unmodelled dynamics or even very small bounded disturbances can cause instability in most of the adaptive control algorithms.

Many efforts to design robust adaptive controllers in the case of unknown parameters have consistently progressed along two different shapes[26-31]. In the first, the adaptive law is altered so that the overall system has bounded solutions in the presence of bounded disturbances. The second relies on the persistent excitation of certain relevant signals in the adaptive loop. The next subsections will present some of the main "modifications" proposed to enforce robustness with bounded disturbances.

2.2. Dead-zone modification

In many physical devices, the output is zero until the magnitude of the input exceeds a certain value. Such an input-output relation is called a dead-zone[32]. In a first approach to prevent instability of the adaptive process in the presence of bounded external disturbances, Egardt[26] introduced a modification of the law so that adaptation takes place when the identification error exceeds a certain threshold. The term dead-zone was first proposed by Narendra and Peterson[18] in 1980, where the adaptation process stops when the norm of the state error vector becomes smaller than a prescribed value. In 1980, the study was initiated by Narendra to determine an adaptive control law that ensures the boundedness of all signals in the presence of bounded disturbances, in the case of continuous systems. In the study of Peterson and Narendra[30], they highlight the cruciality of the proper choice of the dead zone for establishing global stability in the presence of an external disturbance. A larger dead zone implies that adaptation will take place in shorter periods of time, which also means larger parameter errors and larger output. One of the assumptions made in this paper is that a bound of the disturbance can be determined even though the plant parameters are unknown. The adaptive law shall consider that the module of the augmented error is not greater than the bound plus an arbitrary positive constant. Hence, the only knowledge needed to calculate the size of the dead zone is the bound of the disturbance, which can be computed[30]. It is also worth noting that no prior knowledge of the zeros of the plant's transfer function is needed to find the bound.

Samson[28] presented a brief study in 1983 based on all his previous works and the analysis of Egardt in his book. Although his paper was only concerned with the stability analysis and not the convergence of the adaptive control to an optimal state, he was able to efficiently introduce a new attempt to use the possible statistical properties of the bounded disturbances. The three properties P_1-P_3 should be verified by the identification algorithm and are similar to the ones demanded for the disturbance-free cases, but less restrictive. The first property states that the identified vector has to be uniformly bounded, which prevents the system from diverging faster than exponentially. The second property ensures that the prediction error remains relatively small, which indicates that the "adaptive observer" transfer function is very similar to that of the system. Finally, the third property allows the control of the time-varying adaptive observer of the system.

In 1983, a modified dead-zone technique was proposed in Bunich's research[33] and was widely used. This modification permits a size reduction of the residual set for the error, hence, simplifying the convergence proof. The drawback is the necessary, yet restrictive, knowledge of a bound on the disturbance in order to appropriately determine the size of the dead-zone.

The work of Peterson and Narendra invigorated a new study by Sastry[34] where he examined the robustness aspects of MRAC. Sastry used the same approach to show that a suitably chosen dead-zone can also stabilize the adaptive system against the effects of unmodelled dynamics. Though, the error between the plant and the model output does not converge to zero but rather to a magnitude less than the size of the dead-zone. In other terms, no adaptation takes place when the system is unable to distinguish between the error signal and the disturbance.

The issue in the dead-zone modification is that it is not Lipschitz, which may cause high-frequency oscillations and other undesirable effects when the tracking error is at or near the dead-zone boundary. In 1986, Slotine and Coetsee[35] proposed a "smoother" version of the dead-zone modification. Unfortunately, we were not able to get a hold of a copy of this paper, but the major idea was explained in his book in 1990[32].

2.3. σ-modification

The dead-zone modification assumes a priori knowledge of an upper bound for the system disturbance. On the other hand, the σ-modification scheme does not require any prior information about bounds for the disturbances. This modification was proposed by Ioannou and Kokotovic[36] in 1983, which Ioannou referred to later as "fixed σ-modification". The modification basically adds damping to the ideal adaptive law. They introduced the modification by adding a decay term $-\sigma\Gamma\theta$ to the disturbance-free integral adaptive law, where σ is a positive scalar to be chosen by the designer. The stability properties with the modification were established based on the existence of a positive constant p such that, for $\sigma > p$, the solutions for the error and adaptive law equations are bounded for any bounded initial condition. A conservative value of σ has to be chosen in order to guarantee $\sigma > p$. It was also shown that the modification yields the local stability of an MRAC scheme when the plant is a linear system of relative degree one and has unmodeled parasitics.

However, even though the robustness achievement is done smoothly and in a simpler way with the σ-modification scheme, there is thepotential destruction of some of the convergence properties, since there is no more asymptotic convergence and the fine tracking error is confined within a bounded region. Consequently, many additional modifications have been suggested later, motivated by the aforementioned drawback of the σ-modification. In 1986, Ioannou and Tsakalis[37] proposed the "switching σ-modification". In contrast to Ioannou's earlier work[38,39], the switching of σ from 0 to σ_0 is modified so that σ is a continuous function of $|\theta(t)|$ [Equation (16)], since the previous modification choices forced the adaptive law to be discontinuous, which might not guarantee the existence of a solution and would probably cause oscillations on the switching surface during implementation. Hence, the continuous switching, as shown in Figure 2, replaces the discontinuous one and is defined in Equation (16),

$$\sigma_s = \begin{cases} 0 & if\,|\theta(t)| < M_0 \\ \sigma_0\left(\frac{|\theta(t)|}{M_0}\text{-}1\right) & if\,M_0 \le |\theta(t)| \le 2M_0 \\ \sigma_0 & if\,|\theta(t)| > 2M_0 \end{cases} \tag{16}$$

where $M_0 > 0$, $\sigma_0 > 0$ are design constants and M_0 is chosen to be large enough so that $M_0 > |\theta^*|$.

In 1992, Tsakalis[40] employed the σ-modification to target the adaptive control problem of a linear, time-varying SISO plant. The signal boundedness for adaptive laws was guaranteed using the σ-modification, normalization and a sufficient condition. The condition relates the speed and the range of the plant parameter variations with the σ value and simplifies the selection of the design parameters in the adaptive law.

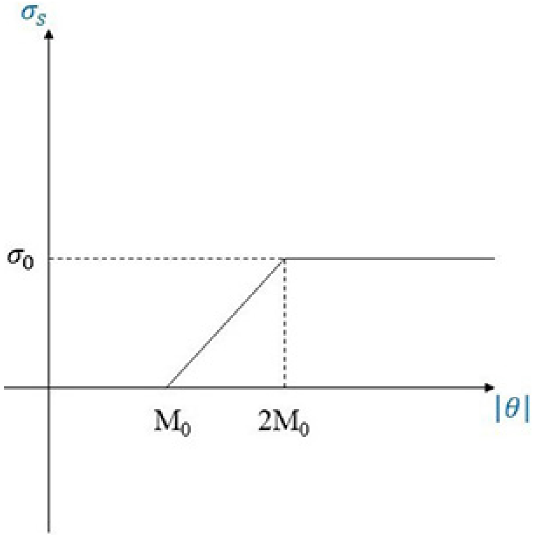


Figure 2. Switching σ-modification (continuous).

In a more recent study, He *et al.*[41] opted to revisit the fundamental σ-modification scheme and propose a qualitative analysis for all the scenarios where this modification can lead to perfect tracking, and where it can allow proper modification of the adaptive laws. The analysis method pre-supposes the existence of a Lyapunov function for an extended system, as shown in the reference[42]. The efficacy of the proposed analysis was demonstrated in a Robust adaptive control system in order to detect its global asymptotic convergence under the fixed σ-modification scheme. When it comes to simulation results, the system shows asymptotic convergence of its trajectories without the modification; however, it may lose its asymptotic stability when the feedback gain and the modification gain are not well designed when using the modification. The recovery of the global asymptotic convergence is primarily dependant on the proper design of both gains, as shown in their last simulation.

2.4. ϵ-modification

The downside of the σ-modification is that when the tracking error becomes small enough, the adaptive parameters have an inclination to revert to the origin, which undoes the gain values that caused the tracking error to become small in the first place. In order to overcome this undesirable effect, Narendra and Annaswamy[43] developed the ϵ-modification. The suggested modification was motivated by that given in the work of Ioannou and Kokotovic[36], which similarly guarantees bounded solutions in the presence of bounded disturbances when the reference input is not persistently exciting, and needs less prior information regarding plant and disturbance. However, the catching point comes when the reference input is persistently exciting and has a sufficiently large amplitude. In this case, as we mentioned earlier, the origin of the error equations is exponentially stable, unlike that in Ioannou's σ-modification. The new adaptive law replaces the σ with a term proportional to the magnitude of the output error, called ϵ (or e_1 in the work of Narendra and Annaswamy[43]).

Ideally, let's consider the first order plant described with Equation (17),

$$\dot{x_p} = a_p x_p + u \tag{17}$$

where a_p is an unknown constant. The reference model is defined in Equation (18),

$$\dot{x_m} = -a_m x_m + r \tag{18}$$

where $a_m > 0$ and r is a bounded piecewise continuous function. Equation (19) shows how the aim of the adaptive control is to choose the control input, u, such that the output of the plant approaches that of the model,

$$u(t) = r(t) + \theta(t) x_p(t) \tag{19}$$

where θ is the control parameter. Therefore, we deduct the error equations in Equation (20),

$$\begin{aligned} \dot{\epsilon} &= -a_m \epsilon + \varphi x_p \\ \varphi &\triangleq \theta(t) - \theta^* \\ \theta^* &\triangleq -a_m - a_p \\ \epsilon &\triangleq x_p - x_m \end{aligned} \tag{20}$$

and hence, based on Equation (21), the proposed adaptive law can be defined,

$$\dot{\varphi} = -\epsilon x_p - \gamma|\epsilon|\theta \qquad \gamma > 0 \tag{21}$$

where ϵ is playing a double role since it attempts to decrease the magnitude of the output error while keeping the parameter θ or the parameter error φ bounded. The choice of the Lyapunov function, in Equation (22), gives the time derivative of V,

$$\dot{V} = -a_m \epsilon^2 - \gamma|\epsilon|\varphi\theta \tag{22}$$

If we define a set *D*,

$$D \triangleq \left\{ (\epsilon,\varphi) | \varphi^2 + \varphi\theta^* + \frac{a_m}{\gamma} |\epsilon| \leq 0 \right\}$$

we then can deduct that $V \leq 0$ inside the set *D*. The modification, which is synthesized by the additional term $-|\epsilon|\theta$ in the adaptive law, shows that the set *D* is compact, which allows us to apply LaSalle's theorem[44] and prove that all solutions of the error equations are bounded.

If we distinguish between the three possible cases based on the reference input states: null, constant or persistently excited, and as mentioned earlier, the third case's application highlights the difference between the proposed modification and the aforementioned σ-modification. When the σ-modification is used to adjust the control parameter θ, in the presence of the disturbance v, we can set the error equations as in Equation (23),

$$\begin{cases} \dot{\epsilon} = -a_m\epsilon^2 + x_p\varphi + v \\ \dot{\varphi} = -\left(x_p\epsilon + \sigma\varphi + \sigma\theta^*\right) \end{cases} \tag{23}$$

where it has been shown that three stable equilibrium states exist in case $x_m = 0$, none of them is the origin, and has a single equilibrium state whose distance from the origin decreases as the amplitude of x_m increases, and that is in case x_m is a constant. Which clearly highlights the addition of the ϵ-modification.

2.5. Summary

Basically, the approaches discussed above reduce the effects of parameter adaptation when the measure error is not due to an error in the parameter estimates. They contribute to either parameter error, noise error, high-frequency unmodelled dynamics error, or disturbances, which consist of anything undescribed by the three previous groups[45]. A brief comparison of all the aforementioned modification techniques is shown in Table 1.

Considering the robustness problem, one can see that the disturbance is generated internally, which makes it dependable in the actual plant's input-output signals. Particularly, the disturbance will grow unboundedly if the adaptive system is unstable and the input-output signals are growing without bound. Videlicet, the stability problem becomes internal and signal dependant. Thus, the boundedness of the disturbance should not be presumed, which proves that, despite the intrinsic results shown in the previous literature, the aforementioned approaches do not necessarily solve the robustness problem in the presence of bounded disturbances[46].

Over the years, the adaptive controllers have proven themselves effective, especially in the process that can be modeled linearly with slowly time-varying parameters relative to the system's dynamics. The 1980s were the peak of theoretical research on this case. On the other hand, many practical examples can be found in these research[8,47-52].

An overview of some practical examples of adaptive control applications in two different fields, thermal exchange and robotics, is given in Table 2[53-57]. We would also like to refer the readers to a very concise survey written by Åström[8] in 1983 for more practical examples of the applications of adaptive control. In addition, adaptive controllers are extremely practical and fruitful when it comes to servo systems that have large disturbances, like load changes, or uncertainties, like frictions, and that have measurable states. The number one practical field in that era was robotics[53-57].

Obviously, adaptive controllers are not the "perfect" solution to all control problems. For instance, they do not provide stability for systems where parameter dynamics are at least the same magnitude as the system's dynamics. The controller robustness can be improved by employing artificial intelligence (AI) techniques, such as fuzzy logic and neural networks[58-60]. Essentially, these methods approximate a nonlinear function and provide a good representation of the nonlinear unknown plant[61], although it is typically used as a model-free controller. The plant is treated as a "black box", with input and output data gathered and trained

Table 1. Stability analysis of each modification technique

Dead-zone modification	***σ-modification***	***ϵ-modification***
• Developed based on adaptation hibernation principle.	• Adding a damping term to the adaptation law: $K = \Gamma_K(\Psi e^T PB - \sigma K)$, where σ > 0	• Adding an error dependent leakage term to the law: $K = -\Gamma_K e^T BP(\Psi - \epsilon K)$, where $\epsilon > 0$
• Stops adaptation when the error touches the boundary of a compact set β_d: $\beta_d = \{(e,\Delta K^T), e \in R^n, \Delta K \in R^{N\times m}$ $\|\|e\|\| \leq e_d\}$	• Takes different forms depending on the choice of sigma	• Reduces the unbounded behavior of the adaptive law
• Adaptation will be disabled once reaches e_d • Stability is guaranteed outside of β_d • The adaptive law is defined in both conditions as:	• The Lyapunov function derivative is negative under some conditions that define a compact set β_σ: $\beta_\sigma = \{(e,\Delta K^T), e \in R^n, \Delta K \in R^{N\times m}$ $\|\|e\|\| \leq e_\sigma \wedge (\|\|\Delta K\|\|_F \leq \Delta K_\sigma)\}$	• Following the same argument as in sigma modification: the Lyapunov function derivative is negative under certain conditions that define a compact set β_ϵ: $\beta_\epsilon = \{(e,\Delta K^T), e \in R^n, \Delta K \in R^{N\times m}$ $\|\|e\|\| \leq e_\epsilon \wedge (\|\|\Delta K\|\|_F \leq \Delta K_\epsilon)\}$
Drawbacks: a prior knowledge about the upper bound of the disturbance is required	• Error UUB is guaranteed and boundedness of all adaptive gains is also guaranteed	• Error UUB is guaranteed and boundedness of all adaptive gains is also guaranteed
	Drawbacks: the damping term addition may not be convenient in some situations	• The upper bound of the set is determined by the upper bound of the disturbance

Table 2. Practical examples of adaptive control implementation

Approach	**Employed by...**
Robotic manipulators	
MRAC	Dubowsky and DesForges[53] (1979) and Nicosia and Tomei[55] (1984)
STAC	Koivo and Guo[56] (1983)
Adaptive algorithm	Dubowsky[54] (1981) and Horowitz and Tomizuka[57] (1986)
Other applications	
MRAC	Harrell *et al.*[49] (1987) and Davidson[47] (2021)
STAC	Davison *et al.*[48] (1980) and Harris and Billings[52] (1981)
Direct AC	Zhang and Tomizuka[50] (1985)
Function Blocks	Lukas and Kaya[51] (1983)

MRACL Model Reference Adaptive Control; STAC: self-tuning adaptive control.

on. The AI framework addresses the plant's model after the training phase, and can handle the plant with practically no need for a mathematical model. It is feasible to build the complete algorithm using AI techniques, or to merge the analytical and AI approaches such that some functions are done analytically and the remainder are performed using AI techniques[62].

3. NEURAL NETWORKS FOR DYNAMIC SYSTEMS

The sophisticated adaptive control techniques that have been created complement computer technology and offer significant potential in the field of applications where systems must be regulated in the face of uncertainty. In the 1980s, there was explosive growth in pure and applied research related to NN. As a result, MLN and RNN have emerged as key components that have shown to be exceptionally effective in pattern recognition and optimization challenges[63-68]. These networks may be thought of as components that can be employed efficiently in complicated nonlinear systems from a system-theoretic standpoint.

The topic of regulating an unknown nonlinear dynamical system has been approached from a variety of perspectives, including direct and indirect adaptive control structures, as well as multiple NN models. Because NN may arbitrarily simulate static and dynamic, highly nonlinear systems, the unknown system is replaced by a NN model with a known structure but a number of unknown parameters and a modeling error component. With regard to the network nonlinearities, the unknown parameters may appear both linearly and nonlinearly, changing the original issue into a nonlinear robust adaptive control problem.

3.1. Neural network and the control of dynamic nonlinear systems

The characteristic of neural networks is that they are quite parallel. They can speed up computations and assist in the solving of issues that need much processing. Since NNs have nonlinear representations and can respond to changes in the environment, they easily reflect physical conditions like industrial processes and their control, whereas precise mathematical models are harder to construct.

One of the few theoretical frameworks for employing NNs for the controllability and stability of dynamical systems has been established by Levin and Narendra[69]. Their research is limited to feedforward MLNs with dynamic BP and nonlinear systems with full state information access. Figure 3 presents the proposed architecture of the NNs. Equation (24) considers a system at a discrete-time index k,

$$x(k+1) = f\big(x(k), u(k)\big) \tag{24}$$

where $x(k) \in \chi \subset \mathrm{R}^n$, $u(k) \in U \subset \mathrm{R}^r$ and $f(0,0) = 0$ so that $x = 0$ is an equilibrium. Conditions are given, in Equation (25), under which the two following NNs can be trained to feedback linearize and stabilize the system.

$$\begin{cases} u = NN_{\Psi}(v,z) \\ z = NN_{\Phi}(x) \end{cases} \qquad (z,v) \in \mathrm{R}^n \times \mathrm{R}^r \tag{25}$$

The results are extended to non-feedback linearizable systems. If the controllability matrix around the origin has a full rank, a methodology and conditions for training a single NN to directly stabilize the system around the origin have been devised. Narendra and Parthasarathy[70] use NNs to create various identification and controller structures. Although the MLNs represent static nonlinear maps and the RNNs represent nonlinear dynamic feedback systems, they suggest that the feedforward MLNs and RNNs are comparable. They describe four network models of varying complexity for identifying and controlling nonlinear dynamical systems using basic examples.

Sontag proposed an article where he tried to explore the capabilities and the ultimate limitations of alternative NN architectures[71]. He suggests that NNs with two hidden layers may be used to stabilize nonlinear systems in general. Intuitively, the conclusion contradicts NNs approximation theories, which claim that single hidden layer NNs are universal approximators. Sontag's solutions are based on the description of the control issue as an inverse kinematics problem rather than an approximation problem.

In 1990, Barto[72] drew an interesting parallel between connectionist learning approaches and those investigated in the well-established field of classical adaptive control. When utilizing NNs to address a parameter estimate problem, the representations are frequently chosen based on how nervous systems represent information. In contrast, in a traditional method, issue representation options are made based on the physics of the problem. As opposed to conventional methods, a connectionist approach is dependent on the structure of the network and the correlation between the connectionist weights. A traditional controller may readily include a priori information; however, in NNs, it is often an input-output connection. In both

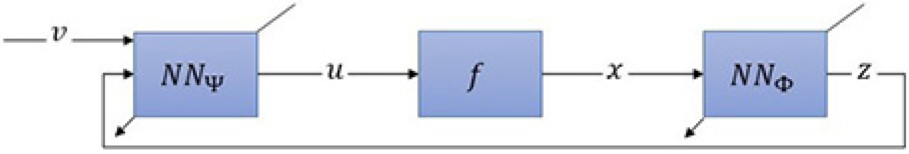


Figure 3. Architecture of the proposed NNs in the work of Levin and Narendra[69]. NNs: Neural Networks.

techniques, performance may be assessed using cost functions such as least mean squared error. All of the training data is available at the same time with off-line approaches. However, with on-line approaches, the required feature is continuous learning, and as a result, the methods must be extremely efficient in order to keep up with the changing events over time.

Adaptive NNs have recently been used by a growing number of academics and researchers to construct acceptable control rules for nonlinear systems. An overview of the primarymost recent literature that implemented adaptive NNs-based techniques is discussed in Table 3[73-82].

3.2. Inverted pendulum

Many researchers have studied learning control using the inverted pendulum problem. The canonical underactuated system, called the cart-pole system, is illustrated in Figure 4. Because deriving the dynamics is relatively simple, it is considered a basic control issue, yet it still hides some underlying complexity owing to its underactuated character. The multiple obstacles that must be addressed to properly regulate such extremely complex nonlinear unstable systems include severe nonlinearities, variable operating circumstances, structured and unstructured dynamical uncertainties, and external disturbances. The purpose of the control is to balance the pole by moving the cart, which has a restricted range of movements. We distinguish between the position of the cart h and its velocity h, and the angle of the pole θ with its angular velocity θ.

In 1983, Barto *et al.*[83] showed how a system consisting of two neuronlike adaptive elements, associative search element (ASE) and adaptive critic element (ACE), can solve a difficult learning control problem such as the cart-pole system. Their work was based on the addition of a single ACE to the ASE developed by Michie and Chambers in the works of Michie and Chambers[84,85]. They have partitioned the state space into 162 boxes. Their simulations revealed that the ASE/ACE system outperformed the boxes system in terms of run time. The system was more likely to solve the problem before it had 100 failures, but the boxes system was less likely to do so. The ASE/ACE system's high performance was nearly completely owing to the ACE's provision of reinforcement throughout the trials. Learning occurs only upon failure with box systems and ASEs without an ACE, which happens less frequently as learning progresses. An ASE can get input on each time step with the ACE in place. The system attempts to access some areas of the state space and avoids others as a result of the learning achieved by this input.

Anderson[86] built on the work of Barto *et al.*[83] by using a variant of the common error BP algorithm to two-layered networks that learn to balance the pendulum given the inverted pendulum's real state variables as input. Two years later[87], he summarized both aforementioned works by discussing the neural network structures and learning methods from a functional viewpoint and by presenting the experimental results. He described NN learning techniques, which use two functions to learn how to construct action sequences. The first is an action function, which converts the current state into control actions. The second is an evaluation function, which converts the present state into an assessment of that state. There were two sorts of networks that emerged: "action and evaluation" networks. This is an adaptive critic architecture version

Table 3. Different adaptive NN-based controls in the recent years

Research	Method/approach	Solved problem
1. Nonaffine nonlinear systems		
Dai *et al.*[73]	Obtaining the implicit desired control input (IDCI), and use of NNs to approximate it	Learning from adaptive NN-based control for a class of nonaffine nonlinear systems in uncertain dynamic environments
Chen *et al.*[74]	The unknown functions are approximated by using the property of the fuzzy-neural control	Adaptive fuzzy-NN (FNN) for a class of nonlinear stochastic systems with unknown functions and a nonaffine pure-feedback form
2. Tracking control		
Dai *et al.*[75]	Radial basis function NNs (RBF-NNs) to learn the unknown dynamics, and adaptive neural control to guarantee the ultimate boundedness (UB)	Stabilization of the tracking control problem of a marine surface vessel with unknown dynamics
Li *et al.*[76]	NNs to approximate the unknown functions, and Barrier Lyapunov function (BLF) for nonstrict-feedback stochastic nonlinear system	Adaptive tracking control for a category of SISO stochastic nonlinear systems with dead zone and output constraint
Cheng *et al.*[77]	Use of NN-based inversion-free controller, and construction of dynamic model using feedforward MLNs	Displacement tracking control of piezo-electric actuators (PEAs)
Ren *et al.*[78]	Use of adaptive neural control, and inclusion of σ-modification to the adaptation law to establish stability	Tracking control problem of unknown nonlinear systems in pure-feedback form with the generalized P-I hysteresis input
3. Unknown model/direction		
Luo *et al.*[79]	Implementing three NNs to approximate the value function, control and disturbance policies, respectively	Date-driven H_∞ control for nonlinear distributed parameter systems with a completely unknown model
Liu *et al.*[80]	Two types of BLFs are used to design the controller and analyze the stability	Stabilize a class of nonlinear systems with the full state constraints and the unknown control direction
4. Backstepping design		
Li *et al.*[81]	Adaptive backstepping control and RBF-NNs.	Overcoming the robustness issues of backstepping design and its uncertainty.
5. Discrete-time systems		
Zhang *et al.*[82]	Iterative adaptive dynamic programming algorithm, with two NNs to approximate the costate function and the corresponding control law	Solving the optimal control problem for discrete-time systems with control constraints

NNs: Neural Networks.

In 1991, Lin and Kim integrated the CMAC into the self-learning control scheme that was based on the work of Lin and Kim[88]. The CMAC model was originally proposed by Albus[89-92] and it was based on models of human memory and neuromuscular control. The CMAC-based technique in the work of Lin and Kim[88] is tested using the inverted pendulum problem, and the results are compared to those of Barto *et al.*[83] and Anderson[87]. The technique has the highest learning speed due to its capability of generalization and good learning behavior. Furthermore, the memory size can be reduced compared to the box-based system. A summarized timeline of the above literature, where NN-based control was implemented to balance the inverted pendulum, is presented in Figure 5.

Many control laws for inverted pendulums have been presented in those research work[93-95], including classical, robust, and adaptive control laws, but they all take structured parametric uncertainty into account. In 2009, Chaoui *et al.*[96] proposed an ANN based adaptive control strategy for inverted pendulums that accomplishes asymptotic motion tracking and posture control with unknown dynamics. Two neural networks ANN_x and ANN_θ are designed to control the motion along the x axis and the pendulum posture with unknown dynamics. Figure 6 shows the block diagram of the proposed system.

Three experiments are carried out to evaluate the performance of the proposed controller. The velocity and posture of the pendulum progressively decrease to zero in the first experiment. The proposed adaptive

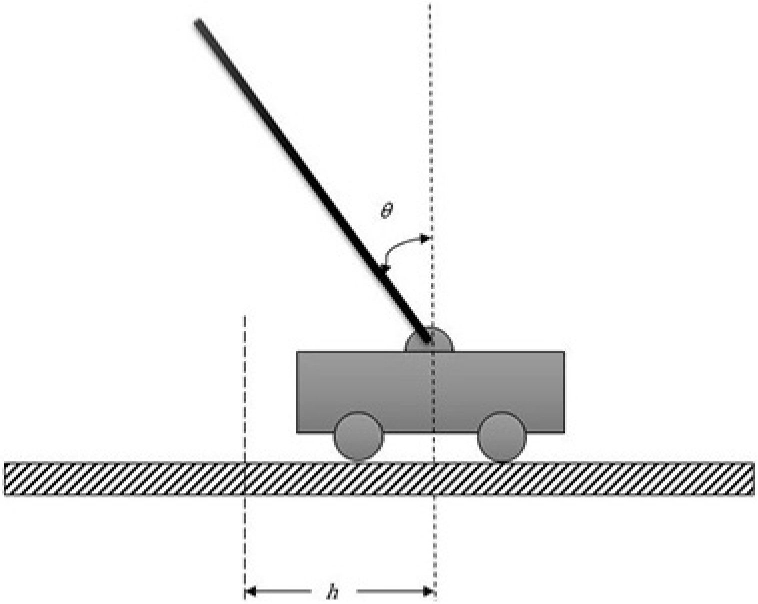


Figure 4. The Cart-pole scheme.

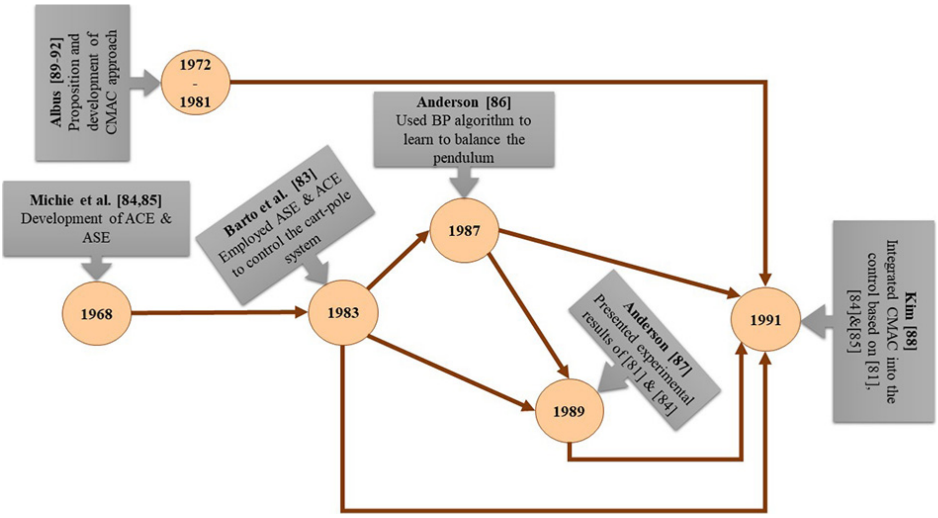

Albus [89-92]
Proposition and development of CMAC approach
1972 - 1981
Anderson [86]
Used BP algorithm to learn to balance the pendulum
Michie et al. [84,85]
Development of ACE & ASE
Barto et al. [83]
Employed ASE & ACE to control the cart-pole system
1968
1983
1987
Anderson [87]
Presented experimental results of [81] & [84]
1989
1991
Kim [88]
Integrated CMAC into the control based on [81], [84]&[85]

Figure 5. Timeline scheme of the works that kickstarted the use of NNs to control the inverted pendulum. NNs: Neural Networks.

control, on the other hand, produces a smooth control signal. The controllers also deal with friction nonlinearities and accomplish quick error convergence and tracking. The second experiment introduces a starting posture position to test the controller's capacity to correct for a non-zero position error. Posture control takes precedence over motion tracking, as posture is critical for such systems. The purpose of the third experiment is to demonstrate the modularity of the proposed controller in terms of adjusting for external disturbances. The suggested controller's design does not clearly model the induced external disturbance, which generally has a considerable impact on the positioning system's accuracy and generates unacceptably high-frequency oscillations. The controller is able to deal with the unexpected force change

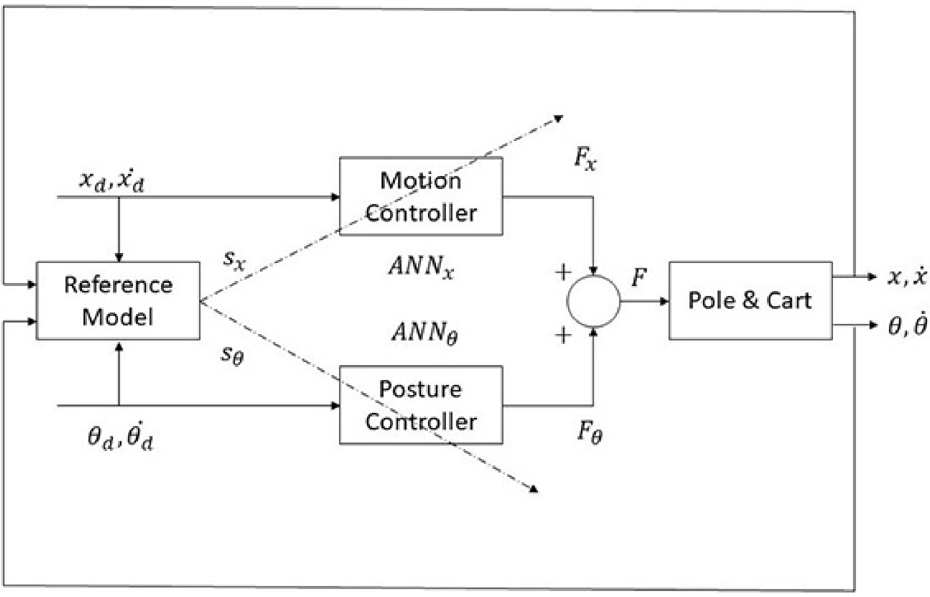


Figure 6. Block diagram of the ANN-based adaptive control scheme[96]. ANN: Artificial Neural Network.

successfully. Furthermore, the motion and posture errors are kept to a minimum, resulting in a smooth control signal.

3.3. Applications for robotic manipulators

There has been great interest in universal controllers that mimic the functions of human processes to learn about the systems they are controlling on-line so that performance improves automatically. NN-based controllers are derived for robot manipulators in a variety of applications, including position control, force control, link flexibility stabilization and the management of high-frequency joint and motor dynamics. The nature of joint torques must be determined for the end effector to follow the required trajectory as quickly and accurately as feasible, which is a common difficulty for robot manipulators. Both parametric and structural uncertainties necessitate adaptive control. Parametric uncertainties originate from a lack of accurate information about the manipulator’s mass characteristics, unknown loads, and load location uncertainty, among other things. Structural uncertainties are a result of the presence of high-frequency unmodeled dynamics, resonant modes, and other structural reservations.

The late 1980s and early 1990s were booming years for both NNs and robotic manipulators research. In this era, the literature survey concerning the application of NNs in robotic manipulators is very rich. Thus, we direct the readers to some interesting approaches in these studies[97-102] and the references therein.

From 1987 to 1989, Miller *et al.*[103-107] discuss a broad CMAC learning technique and its application to robotic manipulators’ dynamic control. The dynamics do not need to be known in this application. Through input and output measurements, the control scheme learns about the process. The findings show that when compared to fixed-gain controllers, the CMAC learning control performs better. Also, because measured and estimated values must be transformed to discrete form, each variable’s resolution and range must be carefully selected, and the number of memory regions handled by each input state in the CMAC architecture is the most important design parameter. In another popular approach, Miller *et al.*[108] used CMAC in the real-time control of an industrial robot and other applications. In their network, they utilize hundreds of thousands of adjustable weights that, in their experience, converge in a few iterations.

Huan *et al.*[109] examine the issue of building robot hand controllers that are device-dependent. Their argument for a controller like this is that it would isolate low-level control issues from high-level capabilities. They employ a BP algorithm with a single hidden layer comprised of four neurons to achieve this goal. The inputs are determined by the object's size, while the outputs are determined by the grab modes. In this way, they have demonstrated how to build a p-g table using simulation. Another BP architecture was used by Wang and Yeh[110] to control a robot model which simulates PUMA560. A network to simulate the plant and a controller network make up their self-adaptive neural controller (SANC). The plant model is trained either off-line with mathematical model outputs or on-line with plant outputs through excitations. The control network is modified by working in series with the plant network during the "controlling and adapting" phase. The control network is also trained off-line in a "memorizing phase" with data from the adapting phase in a random way, which is another element of this training. This trait, according to the authors, aids in overcoming the temporal instability that is inherent with BP. Their numerical findings show that the SANC technique produces good trajectory-tracking accuracy.

Up to the early 2000s, the main goal of robotic manipulators designs was to minimize vibration and achieve good position accuracy, which led to maximizing stiffness. This high stiffness is achieved by using heavy material and a bulky design. As a result, it is demonstrated that heavy rigid manipulators are wasteful in terms of power consumption and operational speed. It is necessary to reduce the weight of the arms and increase their speed of action in order to boost industrial output. As a result of their light weight, low cost, bigger work volume, improved mobility, higher operational speed, power economy, and a wider range of applications, flexible-joint manipulators have gotten much attention. Figure 7 shows a representation of a flexible joint manipulator model.

Controlling such systems, however, still challenges significant nonlinearities, such as coupling caused by the manipulator's flexibility, changing operating conditions, structured and unstructured dynamical uncertainties, and external disturbances. Complex dynamics regulate flexible-joint manipulators[111-114]. This emphasizes the need to examine alternate control techniques for these types of manipulator systems in order to meet their increasingly stringent design criteria. Many control laws for flexible joints have been presented in those studies[115-118] to solely address (structured) parametric uncertainties. The proposed controllers need a complete a priori knowledge of the system dynamics. Several adaptive control systems[119-121] have been proposed to alleviate this necessity. The majority of these control strategies use singular perturbation theory to extend adaptive control theory established for rigid bodies to flexible ones[122-125].

Based on all the above reasons, computational intelligence techniques, such as ANNs and fuzzy logic controllers, have been credited in a variety of applications as powerful controllers of the types of systems that may be subjected to structured and unstructured uncertainties[126,127]. As a result, there have been advancements in the field of intelligent control[128,129]. Various neural network models have been used to operate flexible-joint manipulators, and the results have been adequate[130]. Chaoui *et al.*[131,132] developed a control strategy inspired by sliding mode control that uses a feedforward-NN to learn the system dynamics. Hui *et al.*[133] proposed a time-delay neuro-fuzzy network. The joint velocity signals were estimated using a linear observer in this system, which avoided the need to measure them directly. Subudhi and Morris[134] proposed a hybrid architecture that included a NN for controlling the slow dynamic subsystem and an H_∞ for controlling the rapid dynamic subsystem. Despite its effectiveness, NN-based control systems are still unable to incorporate any humanlike experience already obtained about the dynamics of the system in question, which is regarded as one of the soft computing approaches' primary flaws.

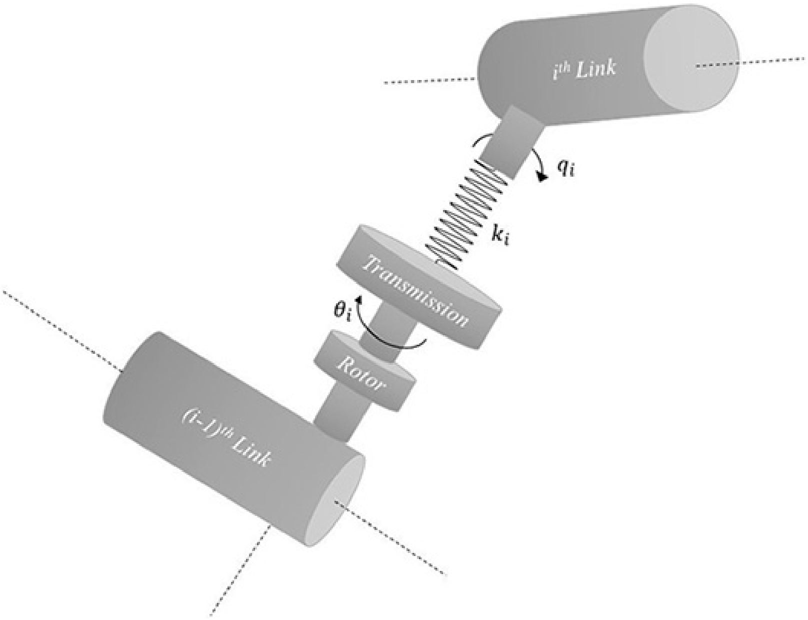


Figure 7. Flexible-joint manipulator model.

Chaoui *et al.*[135] suggested an ANN-based control technique in 2009, which used ANNs' learning and approximation skills to estimate the system dynamics. The MRAC is made up of feedforward (ANN_{FF}) and feedback (ANN_{FBK}) NN-based adaptive controllers. The reference model is built in the same manner as a sliding hyperplane in variable structure control, and its output, which may be regarded as a filtered error signal, is utilized as an error signal to adjust the ANN_{FBK}'s weights. It comprises a first-order model that specifies the required dynamics of the error between the desired and real load locations, as well as between the motor and load velocity, in order to maintain internal stability. The ANN_{FF} offers an approximate inverse model for the positioning system, while the ANN_{FBK} corrects residual errors, assuring the manipulator's internal stability and rapid controller response.

The feedback's learning rate is dependent on the load inertia, which is a flaw in this construction. To improve the stability region of the NN-based controllers, a supervisor is proposed to modify the learning rate of the ANNs. The supervisor also increases the adaptation process's convergence qualities.

Nowadays, the subject of multiple-arms manipulation highlights some interesting progress in using intelligent control approaches. Hou *et al.*[136] used a dual NN to solve a multicriteria optimization problem for coordinated manipulation. Li *et al.*[137,138] are representatives who operate on several mobile manipulators with communication delays. Some promising approaches, such as LMI and fuzzy-NN controls, were used in both articles[137,138], to improve motion/force performances, which were crucial in multilateral teleoperation applications.

In 2017, He *et al.*[139] proposed an Adaptive NN-based controller for a robotic manipulator with time-varying output constraints. The adaptive NNs were utilized to adjust for the robotic manipulator system's uncertain dynamics. The disturbance-observer (DO) is designed to compensate for the influence of an unknown disturbance, and asymmetric barrier Lyapunov Functions (BLFs) are used in the control design process to avoid violating time-varying output constraints. The effects of system uncertainties are successfully corrected, and the system's resilience is increased using the adaptive NN-based controller. The NN estimating errors are coupled with the unknown disturbance from people and the environment to form a combined disturbance that is then approximated by a DO.

In a recent interesting paper, He *et al.*[140] attempted to control the vibrations of a flexible robotic manipulator in the presence of input dead-zone. The lumped technique is used to discretize the flexible link system[141,142]. A weightless linear angular spring and a concentrated point mass are used to partition the flexible link into a finite number of spring-mass parts. They design NN controllers with complete state feedback and output feedback based on the constructed model. All state variables must be known to provide state feedback. An observer is presented to approximate the unknown system state variables in the case of control with output feedback. In summary, an overview of the evolution of NNs implementation in robotic manipulation is shown in Table 4. Each of these papers has been categorized based on the nature of its approach.

3.4. From machine learning to deep learning

ML has transformed various disciplines in the previous several decades, starting in the 1950s. NN is a subfield of ML, a subset of AI, and it is this subfield that gave birth to Deep Learning (DL). There are three types of DL approaches: supervised, semi-supervised, and unsupervised. There is also a category of learning strategy known as RL or DRL, which is commonly considered in the context of semi-supervised or unsupervised learning approaches. Figure 8 shows the classification of all the aforementioned categories.

The common-sense principle behind RL is that if an action is followed by a satisfying state of affairs, or an improvement in the state of affairs, the inclination to produce that action is enhanced, or in other words reinforced. Figure 9 presents a common diagram model of general RL. The origin of RL is well rooted in computer science, though similar methods such as adaptive dynamic programming and neuro-dynamic programming (NDP)[143] were developed in parallel by researchers and many others from the field of optimal control. NDP was nothing but reliance on both concepts of Dynamic-Programming and NN. For the 1990's AI community, NDP was called RL. This is what makes RL one of the major NN approaches to learning control[60].

On the other hand, deep models may be thought of as deep-structured ANNs. ANNs were first proposed in 1947 by Pitts and McCulloch[144]. Many major milestones in perceptrons, BP algorithm, Rectified Linear Unit, Max-pooling, dropout, batch normalization, and other areas of study were achieved in the years that followed. DL's current success is due to all of these ongoing algorithmic advancements, as well as the appearance of large-scale training data and the rapid development of high-performance parallel computing platforms, such as Graphics Processing Units[145]. Figure 10 shows the main types of DL architectures. In 2016, Liu *et al.*[146] proposed a detailed survey about DL architectures. Four main deep learning architectures, which are restricted Boltzmann machines (RBMs), deep belief networks (DBNs), autoencoder (AE), and convolutional neural networks (CNNs), are reviewed.

4. RL/DRL FOR THE CONTROL OF ROBOT MANIPULATION

DRL combines ANN with an RL-based framework to assist software agents in learning how to achieve their

Table 4. NN-based control in robotic manipulation - an overview

Approach	Employed by...
Backpropagation	Elsley[98] (1988), Huan *et al.*[109] (1988), Karakasoglu and Sundareshan[100] (1990) and Wang and Yeh[110] (1990)
CMAC learning	Miller *et al.*[103-108] (1987-1990)
Adaptive NNs/PG table	Huan *et al.*[109] (1988) and He *et al.*[139] (2017)
NNs for flexible joints	Hui *et al.*[133] (2002), Gueaieb *et al.*[128] (2003), Chaoui *et al.*[131,132] (2004), Subudhi and Morris[134] (2006), Chaoui *et al.*[130] (2006), Chaoui and Gueaieb[126] (2008), He *et al.*[140] (2017) and Sun *et al.*[142] (2017)
NNs for multiple arms	Hou *et al.*[136] (2010), Li and Su[137] (2013) and Li *et al.*[138] (2014)
Feedforward and feedback	Chaoui *et al.*[135] (2009)
RNNs	
Hopfield net	Xu *et al.*[101] (1990)
Comparison	Wilhelmsen and Cotter[102] (1990)

NNs: Neural Networks; CMAC: cerebellar model articulation controller; RNNs: recurrent NNs.

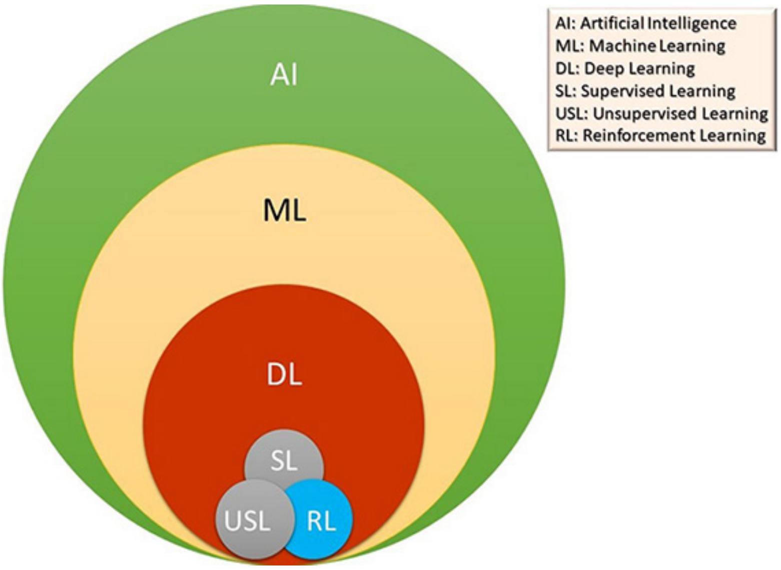

AI
ML
DL
SL
USL
RL
AI: Artificial Intelligence
ML: Machine Learning
DL: Deep Learning
SL: Supervised Learning
USL: Unsupervised Learning
RL: Reinforcement Learning

Figure 8. Classification of AI categories.

objectives. It combines function approximation and goal optimization to map states and actions to the rewards they result in. The combination of NN with RL algorithms led to the creation of astounding breakthroughs like Deepmind's AlphaGo, an algorithm that beat the world champions of the Go board game[147].

As mentioned earlier, RL is a powerful technique for achieving optimal control in robotic systems. Traditional optimal control has the drawback of requiring complete understanding of the system's dynamics. Furthermore, because the design is often done offline, it is unable to deal with the changing dynamics of a system during operation, such as service robots that must execute a variety of duties in an unstructured and dynamic environment. The first chapter of this paper has shown that adaptive control, on

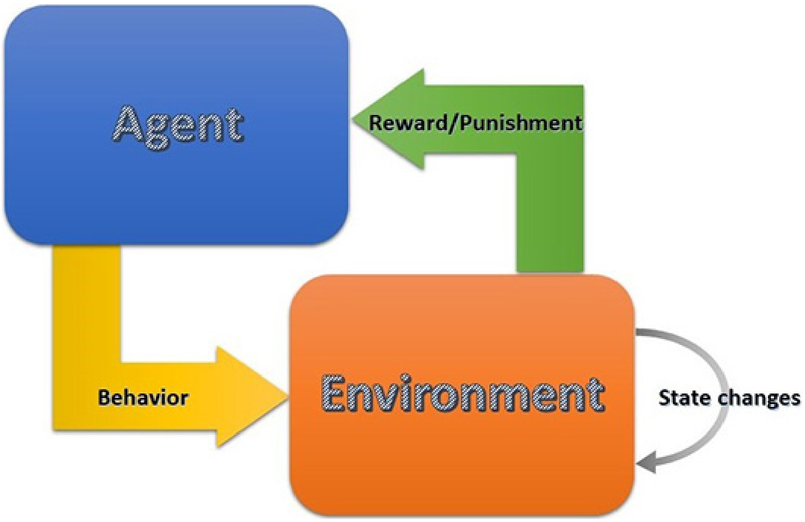


Figure 9. Universal model of RL. RL: Reinforcement learning.

the other hand, is well known for online system identification and control. Adaptive control, on the other hand, is not necessarily optimal and may not be appropriate for applications such as humanoid robots/service robots, where optimality is essential. Furthermore, robots that will be employed in a human setting must be able to learn over time and create the best biomechanical and robotics solutions possible while coping with changing dynamics. Optimality in robotics might be defined as the use of the least amount of energy or the application of the least amount of force to the environment during physical contact. Aspects of safety, such as joint or actuator restrictions, can also be included in the cost function.

4.1. Reinforcement learning for robotic control

The reinforcement learning (RL) domain of robotics differs significantly from the majority of well-studied RL benchmark issues. In robotics, assuming that the true state is totally visible and noise-free is typically impractical. The learning system will have no way of knowing which state it is in, and even very dissimilar states may appear to be quite similar. As a result, RL in robots is frequently represented as a partially observed system. Consequently, the learning system must approximate the real state using filters. Experience with an actual physical system is time-consuming, costly, and difficult to duplicate. Because each trial run is expensive, such applications drive us to concentrate on issues that do not surface as frequently in traditional RL benchmark instances. Appropriate approximations of state, policy, value function, and/or system dynamics must be introduced in order to learn within a tolerable time period. While real-world experience is costly, it can typically not be substituted solely by simulation learning. Even little modeling flaws in analytical or learned models of the system might result in significantly divergent behavior, at least for highly dynamic jobs. As a result, algorithms must be resistant to under-modeling and uncertainty.

Another issue that arises frequently in robotic RL is generating appropriate reward functions. To cope with the expense of real-world experience, rewards that steer the learning system fast to success are required. This problem is known as reward shaping, and it requires a significant amount of manual contribution[148]. In robotics, defining excellent reward functions necessitates a substantial degree of domain expertise and can be difficult in practice.

Not all RL methods are equally appropriate for robotics. Indeed, many of the methods used to solve complex issues thus far have been model-based, and robot learning systems frequently use policy search

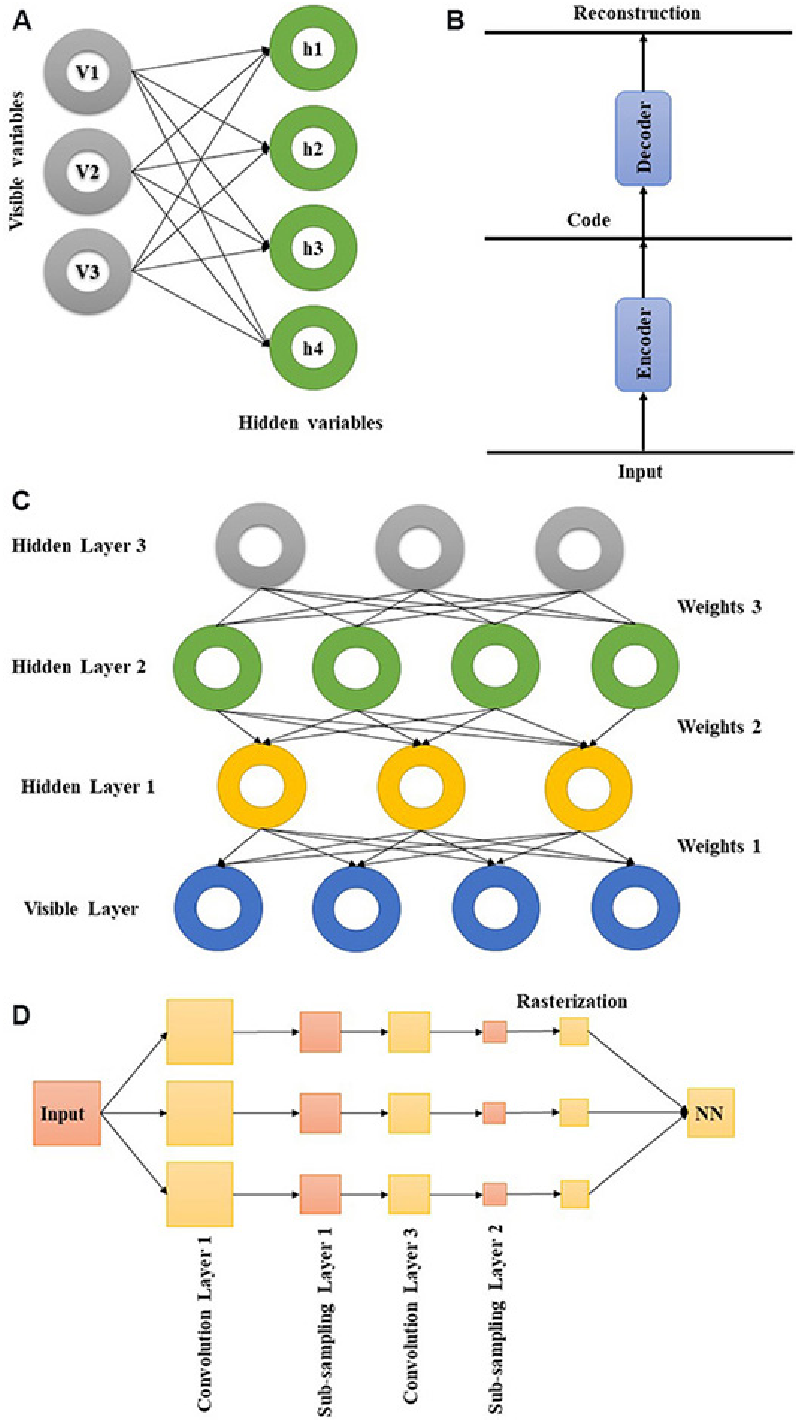


Figure 10. Main DL architectures. (A) Schematic diagram of RBMs; (B) schematic diagram of AEs; (C) schematic diagram of DBNs; (D) conceptual structure of CNNs. RBMs: restricted Boltzmann machines; DBNs: deep belief networks; AE: autoencoder; CNNs: convolutional neural networks.

methods rather than value function-based approaches. Such design decisions are in stark contrast to maybe the majority of early ML research. The papers that follow will discuss several approaches to incorporating

RL into robotics and manipulation. Kober *et al.*[149] conducted a comprehensive review of RL in robotics in 2013. They provide a reasonably comprehensive overview of "Real" Robotic RL and mention the most innovative studies, which are organized by significant findings.

In the last 15 years or so, the use of RL in robots has continuously risen. An overview of the RL-based implementation in robots' control is shown in Table 5[150-172], where each of the undermentioned papers has been categorized based on the nature of their approach.

A stacked Q-learning technique for a robot interacting with its surroundings was introduced by Digney[150]. In an inverted pole-balancing issue, Schaal[151] employed RL for robot learning. For compliance tasks, Kuan and Young[152] developed an RL-based mechanism in conjunction with a robust sliding mode impedance controller, which they evaluated in simulation. To cope with the variation in the different compliance tasks, they apply an RL-based method in their research. Bucak and Zohdy[153,154] proposed an RL-based control strategy for one and two link robots in 1999 and 2001. Althoefer *et al.*[155] used RL to attain motion and avoid obstacles in a Fuzzy rule-based system for a robot manipulator. Q-learning for robot control was investigated by Gaskett[156]. For a mobile robot navigation challenge, Smart and Kaelbling also opted for an RL-based approach[157]. For optimal control of a musculoskeletal-type robot arm with two joints and six muscles, Izawa *et al.*[158] used an RL actor-critic framework. For an optimum reaching task, they employed the proposed technique. RL approaches in humanoid robots are characterized, by Peters *et al.*[159], as greedy methods, "vanilla" policy gradient methods, and natural gradient methods. They highly encourage the adoption of a natural gradient approach to control humanoid robots, because natural-actor-critic (NAC) structures converge fast and are better suited to high-dimensional systems like humanoid robots. They have proposed a number of different ways to design RL-based control systems for humanoid robots. An expansion of this study was given in 2009 by Bhatnagar *et al.*[160]. Theodorou *et al.*[161] employed RL for optimal control of arm kinematics. NAC applications in robotics were presented by Peters and Schaal[162]. For the estimate, the NAC employs the natural gradient approach. Other works presented here[163-165] go into greater depth on actor-critic based RL in robots. Buchli *et al.*[166] propose RL for variable impedance management methods based on policy improvement using a route integral approach. Only simulations were used to illustrate the efficiency of the suggested method. Theodorou *et al.*[167] used a robot dog to evaluate RL based on policy improvement using path integral[168]. RL-based control for robot manipulators in uncertain circumstances was given by Shah and Gopal[169]. Kim *et al.*[170,171] applied an RL-based method to determine acceptable compliance for various scenarios by interaction with the environment. The usefulness of Kim *et al.*[170,171]'s RL-based impedance learning technique has been demonstrated in simulations.

For a robot goalkeeper and inverted pendulum examples, Adam *et al.*[172] proposed a very interesting article on the experimental implementation of experience replay Q-learning and experience replay SARSA approaches. In this form of RL scheme, the data obtained during the online learning process is saved and fed back to the RL system continuously[172]. The results are encouraging, albeit the implementation method may not be appropriate for all actual systems, as the exploration phase indicates very irregular, nearly unstable behavior, which might harm a more delicate plant.

It is worth noting that several of the RL systems outlined above are conceptually well-developed, with convergence proofs available. However, there is still much work to be done on RL, and real-time implementations of most of these systems are still a great difficulty. Furthermore, adequate benchmark challenges[173] are required to test newly created or improved RL algorithms.

Table 5. RL-based control in robotic control - an overview

Approach	Employed by...
Q-learning	Digney[150] (1996), Gaskett[156] (2002), Shah and Gopal[169] (2009) and Adam *et al.*[172] (2012)
Optimal control/bio-mimetic learning	Izawa *et al.*[158] (2002) and Theodorou *et al.*[161] (2007)
NAC	Atkeson and Schaal[163] (1997), Peters *et al.*[159] (2003), Peters and Schaal[162] (2008), Hoffmann *et al.*[164] (2008) and Peters and Schaal[165] (2008)
Inverted pole-balancing	Schaal[151] (1996) and Adam *et al.*[172] (2012)
Impedance control	Kuan and Young[152] (1998) and Buchli *et al.*[166] (2010)
Fuzzy rule-based system	Althoefer *et al.*[155] (2001)
Navigation challenge	Smart and Kaelbling[157] (2002)
Route integral control	Buchli *et al.*[166] (2010)
Path integral	Theodorou *et al.*[167] (2010)

RL: Reinforcement learning; NAC: natural-actor-critic.

4.2. Deep reinforcement learning for robotic manipulation control

In 2012, deep learning (DL) achieved its first major breakthrough with a CNN for classification[174]. It iteratively trains the parameters using loss computation and BP using hundreds of thousands of data-label pairs. Although this approach has developed steadily since its inception and is currently one of the most widely used DL structures, it is not ideal for robotic manipulation control because obtaining a large number of pictures of joint angles with labeled data to train the model is too time-consuming. CNN has been used in several studies to learn the motor torques required to drive a robot using raw RGB video pictures[175]. However, as we will see later, employing deep reinforcement learning (DRL) is a more promising and fascinating notion.

In the context of robotic manipulation control, the purpose of DRL is to train a deep policy NN, such as the one shown in Figure 10, to discover the best command sequence for completing the job. The present state, as shown in Figure 11, is the input, which can comprise the angles of the manipulator's joints, the location of the end effector, and their derivative information, such as velocity and acceleration. Furthermore, the current posture of target objects, as well as the status of relevant sensors if any are present in the surroundings, can be tallied in the current state. The policy network's output is an action that specifies which control instructions, such as torques or velocity commands, should be applied to each actuator. A positive reward will be produced when the robotic manipulator completes a job. The algorithm is supposed to discover the best successful control method for robotic manipulation using these delayed and weak data.

The study of sample efficiency for supervised deep learning determines the scale of the training set required in learning. Consequently, even though it is more challenging than supervised deep learning, the study of sample efficiency for DRL in robotic control provides how much data is needed to build an optimal policy. The first demonstration of using DRL on a robot was in 2015, when Levine *et al.*[176] applied trajectory optimization techniques and policy search methods with NNs to accomplish a practical sample efficient learning. They employ a recently developed policy search approach to learn a variety of dynamic manipulation behaviors with very broad policy representations, without requiring known models or example demonstrations in this study. This method uses repeatedly refitted time-varying linear models to train a collection of trajectories for the desired motion skill, and then unifies these trajectories into a single control policy that can generalize to new scenarios. Some modifications are needed in order to lower the sample count and automate parameter selection to enable this technique to run on a real robot. Finally, this approach has proven that the learning of robust controllers for complexity is possible, which did achieve various compound tasks such as stacking tight-fitting Lego blocks and putting together a toy airplane after

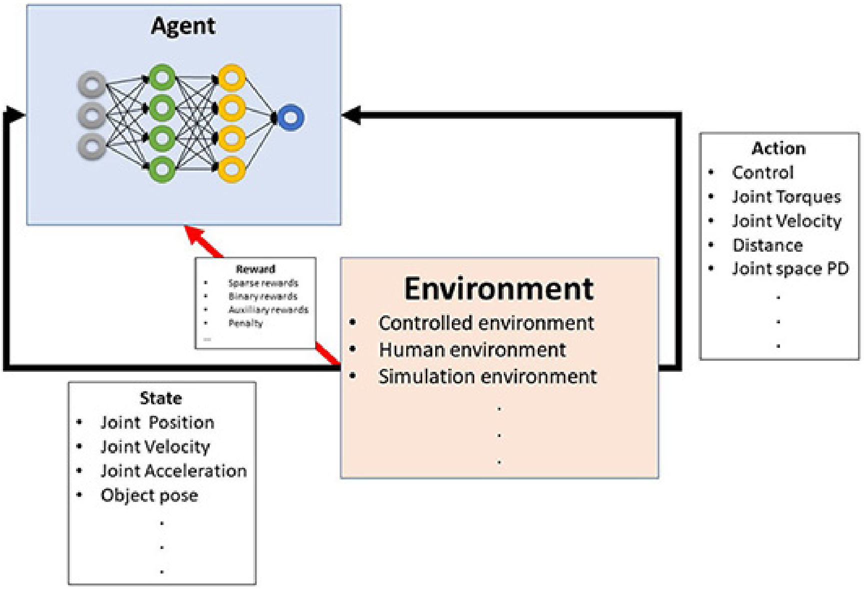


Figure 11. DRL scheme for Robotic manipulation control. DRL: Deep reinforcement learning.

minutes of interaction time.

The concept of imitation learning became very popular for robotic manipulation, since relying on learning from trial and error required a significant amount of system interaction time if based solely on DRL approaches[177]. In 2018, an interesting approach was proposed by Vecerik *et al.*[178] combining both imitation learning and task-reward-based learning, which improved the agent's abilities in simulation. The approach was based on an extension of Deep Deterministic Policy Gradient (DDPG) algorithm for tasks with sparse rewards. Unfortunately, in real robot experiments, the location of the object, as well as the explicit states of joints like position and velocity, must be specified, which limits the approach's applications to high-dimensional data[179].

In 2017, Andrychowicz *et al.*[180] proposed Hindsight Experience Replay as a novel technique that provides for sample-efficient learning from sparse and binary rewards, avoiding the need for complex reward engineering. It may be used in conjunction with any off-policy RL algorithm to create an implicit curriculum.

In October 2021, AI researchers at Stanford University presented a new technique called deep evolutionary reinforcement learning, or DERL[181]. The new method employs a sophisticated virtual environment as well as RL to develop virtual agents that can change their physical form as well as their learning abilities. The discoveries might have far-reaching ramifications for AI research in general and robotics research in particular in the future. Each agent in the DERL architecture employs DRL to gain the abilities it needs to achieve its objectives throughout the course of its existence. MuJoCo, a virtual environment that enables very accurate rigid-body physics modeling, was employed by the researchers to create their framework. Universal Animal is their design space, and the objective is to construct morphologies that can master locomotion and item manipulation tasks in a range of terrains. The developed agents were put through their paces in eight various tasks, including patrolling, fleeing, manipulating items, and exploring. Their findings reveal that AI agents who have developed in different terrains learn and perform better than AI agents who have only seen flat terrain.

An overview of the connection of the above-mentioned work is presented in Table 6. Some basic problems are listed in the table, and each paper's approach is presented and categorized based on observation and action space, reward shaping and algorithm types.

Although DRL-based robotic manipulation control algorithms have proliferated in recent years, the issues of acquiring robust and diverse manipulation abilities for robots using DRL have yet to be properly overcome for real-world applications.

4.3. Summary

Over the last several years, the robotics community has been progressively using RL and DRL-based algorithms to manage complicated robots or multi-robot systems, as well as to give end-to-end policies from perception to control. Since both algorithms base their knowledge acquisition on trial-and-error, they naturally require a large number of episodes, which limits the learning in terms of time and experience variability in real-world scenarios. In addition, the real-world experience must consider the potential dangers or unexpected behaviors of the considered robot, especially when it comes to safety-critical applications. Even though there are some successful real applications to DRL in robotics, especially with tasks involving object manipulations[182,183], the success of its algorithms beyond the simulated worlds is fairly limited. Transferring DRL policies from simulation environments to reality, referred to as "sim-to-real", is a necessary step toward more complex robotic systems that have DL-defined controllers. This has led to an increase in research in "sim-to-real" transfer, which resulted in many publications over the past few years.

Another angle that we see crucial for robotics applications is local *vs.* global learning. For instance, when humans learn a new task, like walking, they automatically build upon the previously learned skill in order to learn a new one, like running, which becomes significantly easier. It is essential to reuse other locally learned information from past data sets. When it comes to robot RL/DRL, the publicity of the making of such data sets with many skills should be available and accessible to everyone in robotic research, which would be considered a huge asset. When it comes to reward shaping, RL approaches have significantly benefited from it by using rewards that convey closeness and are not only based on binary success or failure. For robotics, it is challenging to shape such a reward design, hence, it would be optimal if the reward-shaping is physically motivated, like for instance, minimizing the torques while achieving a task.

5. CONCLUSION

In this review paper, we have surveyed the evolution of adaptive learning for nonlinear dynamic systems. In an initial step, after we introduced adaptive controllers and the modification techniques to overcome bounded disturbances, we have concluded that adaptive controllers have proven their effectiveness, especially in the processes that can be modeled linearly with slowly time-varying parameters relative to the system's dynamics. However, they do not provide stability for systems where parameter dynamics are at least the same magnitude as the system's dynamics.

In an evolutionary manner, AI-based techniques have emerged to improve the controller robustness. Newer methods, such as fuzzy logic and NNs were introduced. Essentially, these methods approximate a nonlinear function and provide a good representation of the nonlinear unknown plant, although it is typically used as a model-free controller. The plant is treated as a "black box", with input and output data gathered and trained on. The AI framework addresses the plant's model after the training phase, and can handle the plant with practically no need for a mathematical model. It is feasible to build the complete algorithm using AI

Table 6. DRL for robotic manipulation categorized by state and action space, algorithm and reward design

State space	Action space	Algorithm type	Reward shaping
Levine *et al.*[176] (2015)			
Joint angles and velocities	Joint torque	Trajectory optimization algorithm.	A penalty term is shaped as the sum of a quadratic term, and a Lorentzian ρ-function The first term encourages speed while the second term encourages precision In addition, a quadratic penalty is applied to joint velocities and torques to smooth and control motions
Andrychowicz *et al.*[180] (2017)			
Joint angles & velocities + Objects' positions, rotations & velocities	4D action space. The first three are position related, the last one specifies the desired distance	HER combined with any off-policy RL algorithm, like DDPG	Binary and sparse rewards
Vecerik *et al.*[178] (2018)			
Joint position and velocity, joint torque, and global pose of the socket and plug	Joint velocities	An off-policy RL algorithm, called DDPGfD, is based on imitation learning	First is a sparse reward function: +10 if the plug is within a small tolerance of the goal The second reward is shaped by two terms: a reaching phase for alignment and an inserting phase to reach the goal
Gupta *et al.*[181] (2021)			
Depends on the agent morphology and include joint angles, angular velocities, readings of a velocimeter, accelerometer, and a gyroscope positioned at the head, and touch sensors attached to the limbs and head	Chosen via a stochastic policy determined by the parameters of a deep NN that are learned via proximal policy optimization (PPO)	DERL, which is a simple computational framework operating by mimicking the intertwined processes of Darwinian evolution	Two reward components. First relative to velocity and second relative to actuators' input

DRL: Deep reinforcement learning; HER: Hindsight Experience Replay; DDPG: Deep Deterministic Policy Gradient.

techniques, or to merge the analytical and AI approaches such that some functions are done analytically and the remainder are performed using AI techniques.

We then briefly presented RL and DRL before we surveyed the previous work implementing both techniques in robot manipulation specifically. From this overview, it was clear that RL and DRL for robotics are not ready to offer a straightforward task yet. Although both techniques have evolved rapidly over the past few years with a wide range of applications, there is still a huge gap between theory and practice. The discrepancy between what we intend to solve and what we solve in practice, and accurately explaining the differences and how this affects our solution, we believe, is one of the core difficulties that plague the RL/DRL research community.

As RL/DRL researchers, we should take a step back and concentrate on the basics. By concentrating on the basics, we imply concentrating on simple, analyzable domains from which we may draw useful conclusions about the algorithms. Above all, areas in which we know what the best possible reward is. We hope that our survey helps the nonlinear dynamic control community in general, and the robotics community in particular, to quickly learn about this topic and become closely familiar with the current work being done and what work remains to be done. We also hope to assist researchers in deriving some conclusions from work carried out so far and provide them with new avenues for future research.

DECLARATIONS

Authors' contributions

Made substantial contributions to the conception and design of the article and interpreting the relevant literature: Harib M

Performed oversight and leadership responsibility for the activity planning and execution, as well as developed ideas and evolution of overarching aims: Chaoui H

Performed critical review, commentary and revision, as well as provided administrative, technical, and material support: Chaoui H, Miah S

Availability of data and materials

Not applicable.

Financial support and sponsorship

None.

Conflicts of interest

All authors declared that there are no conflicts of interest.

Ethical approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

Copyright

REFERENCES

1. Aseltine J, Mancini A, Sarture C. A survey of adaptive control systems. *IRE Trans Automat Contr* 1958;6:102-8. DOI
2. Stromer PR. Adaptive or self-optimizing control systems - a bibliography. *IRE Trans Automat Contr* 1959;AC-4:65-8. DOI
3. Mishkin E, Ludwig BJ. Adaptive control systems. 1st ed. New York: McGraw-Hill; 1961.
4. Truxal JG. Adaptive control. *IFAC Proceedings Volumes* 1963;1:386-92. DOI
5. Eveleigh VW. Adaptive control and optimization technique. 1st ed. New York: McGraw-Hill; 1967.
6. Wittenmark B. Stochastic adaptive control methods: a survey. *Int J Control* 1975;21:705-30. DOI
7. Åström K, Borisson U, Ljung L, Wittenmark B. Theory and applications of self-tuning regulators. *Automatica* 1977;13:457-76. DOI
8. Åström K. Theory and applications of adaptive control - a survey. *Automatica* 1983;19:471-86. DOI
9. Jamali H. Adaptive control methods for mechanical manipulators: a comparative study. Monterey, CA: Naval Postgraduate School; 1989.
10. Mathelin MD, Lozano R. Robust adaptive identification of slowly time-varying parameters with bounded disturbances. *Automatica* 1999;35:1291-305. DOI
11. Deisenroth MP, Rasmussen CE. PILCO: a model-based and data-efficient approach to policy search. Proceedings of the 28th International Conference on International Conference on Machine Learning; 2011 Jun; Madison, WI, USA. 2011. p. 465-72.
12. Wang LY, Zhang JF. Fundamental limitations and differences of robust and adaptive control. Proceedings of the 2001 American

Control Conference. (Cat. No.01CH37148); 2001 Jun 25-27; Arlington, VA, USA. IEEE; 2001. p. 4802-7. DOI

13. Ioannou PA, Sun J. Robust adaptive control. Mineola, NY: Courier Corporation; 2012.
14. Lavretsky E. Adaptive output feedback design using asymptotic properties of LQG/LTR controllers. *IEEE Trans Automat Contr* 2012;57:1587-91. DOI
15. Sastry S, Bodson M. adaptive control: stability, convergence and robustness. Mineola, NY: Dover Publications; 2011. DOI
16. Larminat P. On overall stability of certain adaptive control systems. *IFAC Proceedings Volumes* 1979;12:1153-9. DOI
17. Narendra K, Yuan-hao Lin. Stable discrete adaptive control. *IEEE Trans Automat Contr* 1980;25:456-61. DOI
18. Peterson B, Narendra K. Bounded error adaptive control. *IEEE Trans Automat Contr* 1982;27:1161-8. DOI
19. Fuchs J. Discrete adaptive control: a sufficient condition for stability and applications. *IEEE Trans Automat Contr* 1980;25:940-6. DOI
20. Goodwin G, Ramadge P, Caines P. Discrete-time multivariable adaptive control. *IEEE Trans Automat Contr* 1980;25:449-56. DOI PubMed
21. Egardt B. Global stability analysis of adaptive control systems with disturbances. Proceedings of the 1980 Joint Automatic Control Conference; 2021 Nov 1; San Fransisco, CA. 1980. DOI
22. Rohrs CE, Valavani L, Athans M, Stein G. Robustness of adaptive control algorithms in the presence of unmodeled dynamics. 1982 21st IEEE Conference on Decision and Control; 1982 Dec 8-10; Orlando, FL, USA. IEEE; 1982. p. 3-11. DOI
23. Aström KJ. Analysis of Rohrs counterexamples to adaptive control. The 22nd IEEE Conference on Decision and Control; 1983 Dec; San Antonio, TX, USA. 1983. p. 982-7. DOI
24. Riedle B, Cyr B, Kokotovic P. Disturbance instabilities in an adaptive system. *IEEE Trans Automat Contr* 1984;29:822-4. DOI
25. Ioannou P, Kokotovic P. Instability analysis and improvement of robustness of adaptive control. *Automatica* 1984;20:583-94. DOI
26. Egardt B. Stability of adaptive controllers. Berlin Heidelberg: Springer; 1979. DOI
27. Kreisselmeier G, Narendra K. Stable model reference adaptive control in the presence of bounded disturbances. *IEEE Trans Automat Contr* 1982;27:1169-75. DOI
28. Samson C. Stability analysis of adaptively controlled systems subject to bounded disturbances. *Automatica* 1983;19:81-6. DOI
29. Ioannou PA, Kokotovic PV. Adaptive systems with reduced models. New York, NY, USA: Springer-Verlag; 1983. DOI
30. Peterson B, Narendra K. Bounded error adaptive control. *IEEE Trans Automat Contr* 1982;27:1161-8. DOI
31. Narendra K, Annaswamy A. Robust adaptive control in the presence of bounded disturbances. *IEEE Trans Automat Contr* 1986;31:306-15. DOI
32. Slotine JJE, Li W. Applied nonlinear control. Englewood Cliffs, NJ: Prentice Hall; 1991.
33. Bunich AL. Rapidly converging algorithm for the identification of a linear system with limited noise. *Autom Remote Control* 1983;44:1049-54.
34. Sastry SS. Model-reference adaptive control - stability, parameter convergence, and robustness. *IMA J Math Control Info* 1984;1:27-66. DOI
35. Slotine JE, Coetsee JA. Adaptive sliding controller synthesis for non-linear systems. *International Journal of Control* 1986;43:1631-51. DOI
36. . Adaptive control in the presence of disturbances. In: Ioannou PA, Kokotovic PV, editors. Adaptive systems with reduced models. Berlin/Heidelberg: Springer-Verlag; 1983. p. 81-90. DOI
37. Ioannou P, Tsakalis K. A robust direct adaptive controller. *IEEE Trans Automat Contr* 1986;31:1033-43. DOI
38. Ioannou P. Robust adaptive controller with zero residual tracking errors. *IEEE Trans Automat Contr* 1986;31:773-6. DOI
39. Ioannou P. Robust direct adaptive control. The 23rd IEEE Conference on Decision and Control; 1984 Dec 12-14; Las Vegas, NV, USA. IEEE; 1984. p. 1015-20. DOI
40. Tsakalis KS. The σ-modification in the adaptive control of linear time-varying plants. [1992] Proceedings of the 31st IEEE Conference on Decision and Control; 1992 Dec 16-18; Tucson, AZ, USA. IEEE; 1992. p. 694-8. DOI
41. He Z, Huang D, Xu J. On the asymptotic property analysis for a class of adaptive control systems with σ-modification: adaptive control systems with σ-modification. *Int J Adapt Control Signal Process* 2013;27:620-34. DOI
42. Li MY, Muldowney JS. A geometric approach to global-stability problems. *SIAM Journal on Mathematical Analysis* 1996;27:14. DOI
43. Narendra K, Annaswamy A. A new adaptive law for robust adaptation without persistent excitation. *IEEE Trans Automat Contr* 1987;32:134-45. DOI
44. Lasalle J. Some extensions of Liapunov's second method. *IRE Trans Circuit Theory* 1960;7:520-7. DOI
45. Mattern DL. Practical applications and limitations of adaptive control. Available from: http://www.proquest.com/docview/303617884/abstract/FC4A275C8474474PQ/1 [Last accessed on 8 Mar 2022].
46. Kreisselmeier G, Anderson B. Robust model reference adaptive control. *IEEE Trans Automat Contr* 1986;31:127-33. DOI PubMed PMC
47. Davidson JM. Model reference adaptive control specification for a steam heated finned tube heat exchanger. Available from: https://www.proquest.com/docview/302770965/citation/9192D8E407D24AFBPQ/1 [Last accessed on 8 Mar 2022].
48. Davison E, Taylor P, Wright J. On the application of tuning regulators to control a commercial heat exchanger. *IEEE Trans Automat Contr* 1980;25:361-75. DOI
49. Harrell RC, Kranzler GA, Hsu CS. Adaptive control of the fluid heat exchange process. *J Dyn Syst Meas Control* 1987;109:49-52.

DOI
50. Zhang Q, Tomizuka M. Multivariable direct adaptive control of thermal mixing processes. *J Dyn Syst Meas Control* 1985;107:278-83. DOI
51. Lukas MP, Kaya A. Adaptive control of a heat exchanger using function blocks. *Chemical Engineering Communications* 2007;24:259-73. DOI
52. Harris CJ, Billings SA. Self-tuning and adaptive control - theory and applications. 1st ed. London: Peter Peregrinus, Ltd; 1981.
53. Dubowsky S, Desforges DT. The application of model-referenced adaptive control to robotic manipulators. *J Dyn Syst Meas Control* 1979;101:193-200. DOI
54. Dubowsky S. On the adaptive control of robotic manipulators: the discrete-time case. *IEEE Trans Automat Contr* 1981. DOI
55. Nicosia S, Tomei P. Model reference adaptive control algorithms for industrial robots. *Automatica* 1984;20:635-44. DOI
56. Koivo A, Guo TH. Adaptive linear controller for robotic manipulators. *IEEE Trans Automat Contr* 1983;28:162-71. DOI
57. Horowitz R, Tomizuka M. An adaptive control scheme for mechanical manipulators - compensation of nonlinearity and decoupling control. *J Dyn Syst Meas Control* 1986;108:127-35. DOI
58. Narendra KS, Parthasarathy K. Adaptive identification and control of dynamical systems using neural networks. Proceedings of the 28th IEEE Conference on Decision and Control; 1989 Dec 13-15; Tampa, FL, USA. 1989. p. 1737-8. DOI
59. Lee C. Fuzzy logic in control systems: fuzzy logic controller. II. *IEEE Trans Syst, Man, Cybern* 1990;20:419-35. DOI
60. Sutton RS, Barto AG, Williams RJ. Reinforcement learning is direct adaptive optimal control. *IEEE Control Syst* 1992;12:19-22. DOI
61. Yechiel O. A survey of adaptive control. *IRATJ* 2017;3:0053. DOI PubMed
62. Malik O. Amalgamation of adaptive control and AI techniques: applications to generator excitation control. *Annu Rev Control* 2004;28:97-106. DOI
63. Hopfield JJ. Neural networks and physical systems with emergent collective computational abilities. *Proc Natl Acad Sci U S A* 1982;79:2554-8. DOI PubMed PMC
64. Hopfield JJ, Tank DW. “Neural” computation of decisions in optimization problems. *Biol Cybern* 1985;52:141-52. DOI PubMed
65. Burr D. Experiments on neural net recognition of spoken and written text. *IEEE Trans Acoust, Speech, Signal Processing* 1988;36:1162-8. DOI
66. Gorman R, Sejnowski T. Learned classification of sonar targets using a massively parallel network. *IEEE Trans Acoust, Speech, Signal Processing* 1988;36:1135-40. DOI
67. Sejnowski T, Rosenberg CR. Parallel networks that learn to pronounce English text. *Complex Syst* 1987;1:145-68.
68. Widrow B, Winter R, Baxter R. Layered neural nets for pattern recognition. *IEEE Trans Acoust, Speech, Signal Processing* 1988;36:1109-18. DOI
69. Levin AU, Narendra KS. Control of nonlinear dynamical systems using neural networks: controllability and stabilization. *IEEE Trans Neural Netw* 1993;4:192-206. DOI PubMed
70. Narendra KS, Parthasarathy K. Identification and control of dynamical systems using neural networks. *IEEE Trans Neural Netw* 1990;1:4-27. DOI PubMed
71. Sontag ED. Feedback stabilization using two-hidden-layer nets. *IEEE Trans Neural Netw* 1992;3:981-90. DOI PubMed
72. Barto AG. Connectionist learning for control: an overview. In: Miller WT, Sutton RS, Werbos PJ. Neural networks for control. Cambridge, MA, USA: MIT Press; 1990. p. 5-58.
73. Dai SL, Wang C, Wang M. Dynamic learning from adaptive neural network control of a class of nonaffine nonlinear systems. *IEEE Trans Neural Netw Learn Syst* 2014;25:111-23. DOI PubMed
74. Chen CL, Liu YJ, Wen GX. Fuzzy neural network-based adaptive control for a class of uncertain nonlinear stochastic systems. *IEEE Trans Cybern* 2014;44:583-93. DOI PubMed
75. Dai S, Wang M, Wang C. Neural learning control of marine surface vessels with guaranteed transient tracking performance. *IEEE Trans Ind Electron* 2016;63:1717-27. DOI
76. Li H, Bai L, Wang L, Zhou Q, Wang H. Adaptive neural control of uncertain nonstrict-feedback stochastic nonlinear systems with output constraint and unknown dead zone. *IEEE Trans Syst Man Cybern, Syst* 2017;47:2048-59. DOI
77. Cheng L, Liu W, Hou Z, Yu J, Tan M. Neural-network-based nonlinear model predictive control for piezoelectric actuators. *IEEE Trans Ind Electron* 2015;62:7717-27. DOI
78. Ren B, Ge SS, Su CY, Lee TH. Adaptive neural control for a class of uncertain nonlinear systems in pure-feedback form with hysteresis input. *IEEE Trans Syst Man Cybern B Cybern* 2009;39:431-43. DOI PubMed
79. Luo B, Huang T, Wu HN, Yang X. Data-driven H∞ control for nonlinear distributed parameter systems. *IEEE Trans Neural Netw Learn Syst* 2015;26:2949-61. DOI PubMed
80. Liu Y, Tong S. Barrier Lyapunov functions for Nussbaum gain adaptive control of full state constrained nonlinear systems. *Automatica* 2017;76:143-52. DOI
81. Li Y, Qiang S, Zhuang X, Kaynak O. Robust and adaptive backstepping control for nonlinear systems using RBF neural networks. *IEEE Trans Neural Netw* 2004;15:693-701. DOI PubMed
82. Zhang H, Luo Y, Liu D. Neural-network-based near-optimal control for a class of discrete-time affine nonlinear systems with control constraints. *IEEE Trans Neural Netw* 2009;20:1490-503. DOI PubMed
83. Barto AG, Sutton RS, Anderson CW. Neuronlike adaptive elements that can solve difficult learning control problems. *IEEE Trans*

Syst, Man, Cybern 1983;SMC-13:834-46. DOI
84. Michie D, Chambers RA. Boxes: an experiment in adaptive control. Edinburgh, UK: Oliver and Boyd; 1968. p. 137-52.
85. Michie D, Chambers RA. Boxes' as a model of pattern-formation. 1st ed. Edinburgh: Edinburgh univ. press; 1968. p. 206-15. DOI
86. Anderson CW. Strategy Learning with multilayer connectionist representations. proceedings of the fourth international workshop on machine learning. Elsevier; 1987. p. 103-14. DOI
87. Anderson C. Learning to control an inverted pendulum using neural networks. *IEEE Control Syst Mag* 1989;9:31-7. DOI
88. Lin CS, Kim H. CMAC-based adaptive critic self-learning control. *IEEE Trans Neural Netw* 1991;2:530-3. DOI PubMed
89. Albus JS. Theoretical and experimental aspects of a Cerebellar Model. Available from: https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=820153 [Last accessed on 8 Mar 2022].
90. Albus JS. A new approach to manipulator control: the cerebellar model articulation controller (CMAC). *J Dyn Syst Meas Control* 1975;97:220-7. DOI
91. Albus JS. Mechanisms of planning and problem solving in the brain. *Math Biosci* 1979;45:247-93. DOI
92. Albus JS. Brains, behavior, and robotics. 1st ed. Peterborough: BYTE Books; 1981.
93. Huang, Chien-lo Huang. Control of an inverted pendulum using grey prediction model. *IEEE Trans on Ind Applicat* 2000;36:452-8. DOI
94. Pathak K, Franch J, Agrawal S. Velocity and position control of a wheeled inverted pendulum by partial feedback linearization. *IEEE Trans Robot* 2005;21:505-13. DOI
95. Li, Jun Luo. Adaptive Robust dynamic balance and motion controls of mobile wheeled inverted pendulums. *IEEE Trans Contr Syst Technol* 2009;17:233-41. DOI
96. Chaoui H, Gueaieb W, Yagoub MCE. ANN-based adaptive motion and posture control of an inverted pendulum with unknown dynamics. 2009 3rd International Conference on Signals, Circuits and Systems (SCS); 2009 Nov 6-8; Medenine, Tunisia. IEEE; 2009. p. 1-6. DOI
97. Guez A, Ahmad Z. Solution to the inverse kinematics problem in robotics by neural networks. IEEE 1988 International Conference on Neural Networks; 1988 Jul 24-27; San Diego, CA, USA. IEEE; 1988. p. 617-24. DOI
98. . Elsley. A learning architecture for control based on back-propagation neural networks. IEEE 1988 International Conference on Neural Networks; 1988 Jul 24-27; San Diego, CA, USA. IEEE; 1988. p. 587-94. DOI
99. Jamshidi M, Horne B, Vadiee N. A neural network-based controller for a two-link robot. 29th IEEE Conference on Decision and Control; 1990 Dec 5-7; Honolulu, HI, USA. IEEE; 1990. p. 3256-7. DOI
100. Karakasoglu A, Sundareshan MK. Decentralized variable structure control of robotic manipulators: neural computational algorithms. 29th IEEE Conference on Decision and Control; 1990 Dec 5-7; Honolulu, HI, USA. IEEE; 1990. p. 3258-9. DOI
101. Xu G, Scherrer H, Schweitzer G. Application of neural networks on robot grippers. 1990 IJCNN International Joint Conference on Neural Networks; 1990 Jun 17-21; San Diego, CA, USA. IEEE; 1990. p. 337-42. DOI
102. Wilhelmsen K, Cotter N. Neural network based controllers for a single-degree-of-freedom robotic arm. 1990 IJCNN International Joint Conference on Neural Networks; 1990 Jun 17-21; San Diego, CA, USA. IEEE; 1990. p. 407-13. DOI
103. Miller WT, Glanz FH, Kraft LG. Application of a general learning algorithm to the control of robotic manipulators. *Int J Rob Res* 1987;6:84-98. DOI
104. Miller W. Sensor-based control of robotic manipulators using a general learning algorithm. *IEEE J Robot Automat* 1987;3:157-65. DOI
105. Miller WT. Real time learned sensor processing and motor control for a robot with vision. *Neural Networks* 1988;1:347. DOI
106. Miller WT, Hewes RP. Real time experiments in neural network based learning control during high speed nonrepetitive robotic operations. Proceedings IEEE International Symposium on Intelligent Control 1988; 1988 Aug 24-26; Arlington, VA, USA. IEEE; 1988. p. 513-8. DOI
107. Miller W. Real-time application of neural networks for sensor-based control of robots with vision. *IEEE Trans Syst, Man, Cybern* 1989;19:825-31. DOI
108. Miller W, Glanz F, Kraft L. CMAC: an associative neural network alternative to backpropagation. *Proc IEEE* 1990;78:1561-7. DOI
109. Huan L, Iberall, Bekey. Building a generic architecture for robot hand control. IEEE 1988 International Conference on Neural Networks; 1988 Jul 24-27; San Diego, CA, USA. IEEE; 1988. p. 567-74. DOI
110. Wang SD, Yeh HMS. Self-adaptive neural architectures for control applications. 1990 IJCNN International Joint Conference on Neural Networks; 1990 Jun 17-21; San Diego, CA, USA. IEEE; 1990. p. 309-14. DOI
111. Seidl D, Lam SL, Putman J, Lorenz R. Neural network compensation of gear backlash hysteresis in position-controlled mechanisms. *IEEE Trans on Ind Applicat* 1995;31:1475-83. DOI
112. Olsson H, Åström K, Canudas de Wit C, Gäfvert M, Lischinsky P. Friction models and friction compensation. *European Journal of Control* 1998;4:176-95. DOI
113. Katsura S, Suzuki J, Ohnishi K. Pushing operation by flexible manipulator taking environmental information into account. *IEEE Trans Ind Electron* 2006;53:1688-97. DOI
114. Katsura S, Ohnishi K. Force servoing by flexible manipulator based on resonance ratio control. *IEEE Trans Ind Electron* 2007;54:539-47. DOI
115. Ghorbel F, Hung J, Spong M. Adaptive control of flexible-joint manipulators. *IEEE Control Syst Mag* 1989;9:9-13. DOI
116. Chien M, Huang A. Adaptive control for flexible-Joint electrically driven robot with time-varying uncertainties. *IEEE Trans Ind*

Electron 2007;54:1032-8. DOI

117. Hauschild JP, Heppler GR. Control of harmonic drive motor actuated flexible linkages. Proceedings 2007 IEEE International Conference on Robotics and Automation; 2007 Apr 10-14; Rome, Italy. IEEE; 2007. p. 3451-6. DOI
118. Kong K, Tomizuka M, Moon H, Hwang B, Jeon D. Mechanical design and impedance compensation of SUBAR (Sogang University's Biomedical Assist Robot). 2008 IEEE/ASME International Conference on Advanced Intelligent Mechatronics; 2008 Jul 2-5; Xi'an, China. IEEE; 2008. p. 377-82. DOI
119. Ghorbel F, Spong MW. Adaptive integral manifold control of flexible joint robot manipulators. Proceedings 1992 IEEE International Conference on Robotics and Automation; 1992 May 12-14; Nice, France. IEEE; 1992. p. 707-14. DOI
120. Al-ashoor R, Patel R, Khorasani K. Robust adaptive controller design and stability analysis for flexible-joint manipulators. *IEEE Trans Syst, Man, Cybern* 1993;23:589-602. DOI
121. Ott C, Albu-Schaffer A, Hirzinger G. Comparison of adaptive and nonadaptive tracking control laws for a flexible joint manipulator. IEEE/RSJ International Conference on Intelligent Robots and Systems; 2002 Sep 30-Oct 4; Lausanne, Switzerland. IEEE; 2002. p. 2018-24. DOI
122. Spong MW. Modeling and control of elastic joint robots. *J Dyn Syst Meas Control* 1987;109:310-8. DOI
123. Ge SS, Postlethwaite I. Adaptive neural network controller design for flexible joint robots using singular perturbation technique. *Transactions of the Institute of Measurement and Control* 1995;17:120-31. DOI
124. Taghirad HD, Khosravi MA. Design and simulation of robust composite controllers for flexible joint robots. 2003 IEEE International Conference on Robotics and Automation (Cat. No.03CH37422); 2003 Sep 14-19; Taipei, Taiwan. IEEE; 2003. p. 3108-13. DOI
125. Huang L, Ge SS, Lee TH. Adaptive position/force control of an uncertain constrained flexible joint robots - singular perturbation approach. SICE 2004 Annual Conference; 2004 Aug 4-6; Sapporo, Japan; 2004. p. 220-5.
126. Chaoui H, Gueaieb W. Type-2 fuzzy logic control of a flexible-joint manipulator. *J Intell Robot Syst* 2008;51:159-86. DOI PubMed
127. Karray F, Gueaieb W, Al-Sharhan S. The hierarchical expert tuning of PID controllers using tools of soft computing. *IEEE Trans Syst Man Cybern B Cybern* 2002;32:77-90. DOI PubMed
128. Gueaieb W, Karray F, Al-sharhan S. A robust adaptive fuzzy position/force control scheme for cooperative manipulators. *IEEE Trans Contr Syst Technol* 2003;11:516-28. DOI
129. Kim E. Output feedback tracking control of robot manipulators with model uncertainty via adaptive fuzzy logic. *IEEE Trans Fuzzy Syst* 2004;12:368-78. DOI
130. Chaoui H, Gueaieb W, Yagoub MCE, Sicard P. Hybrid neural fuzzy sliding mode control of flexible-joint manipulators with unknown dynamics. IECON 2006 - 32nd Annual Conference on IEEE Industrial Electronics; 2006 Nov 6-10; Paris, France. IEEE; 2006. p. 4082-7. DOI
131. Chaoui H, Sicard P, Lakhsasi A. Reference model supervisory loop for neural network based adaptive control of a flexible joint with hard nonlinearities. Canadian Conference on Electrical and Computer Engineering 2004 (IEEE Cat. No.04CH37513); 2004 May 2-5; Niagara Falls, ON, Canada. IEEE; 2004. p. 2029-34. DOI
132. Chaoui H, Sicard P, Lakhsasi A, Schwartz H. Neural network based model reference adaptive control structure for a flexible joint with hard nonlinearities. 2004 IEEE International Symposium on Industrial Electronics; 2004 May 4-7; Ajaccio, France. IEEE; 2004. p. 271-6. DOI
133. Hui, Fuchun S, Zengqi S. Observer-based adaptive controller design of flexible manipulators using time-delay neuro-fuzzy networks. *J Intell Robot Syst* 2002;34:453-66. DOI
134. Subudhi B, Morris AS. Singular perturbation based neuro-H_∞ control scheme for a manipulator with flexible links and joints. *Robotica* 2006;24:151-61. DOI
135. Chaoui H, Sicard P, Gueaieb W. ANN-based adaptive control of robotic manipulators with friction and joint elasticity. *IEEE Trans Ind Electron* 2009;56:3174-87. DOI
136. Hou ZG, Cheng L, Tan M. Multicriteria optimization for coordination of redundant robots using a dual neural network. *IEEE Trans Syst Man Cybern B Cybern* 2010;40:1075-87. DOI PubMed
137. Li Z, Su CY. Neural-adaptive control of single-master-multiple-slaves teleoperation for coordinated multiple mobile manipulators with time-varying communication delays and input uncertainties. *IEEE Trans Neural Netw Learn Syst* 2013;24:1400-13. DOI PubMed
138. Li Z, Xia Y, Sun F. Adaptive fuzzy control for multilateral cooperative teleoperation of multiple robotic manipulators under random network-induced delays. *IEEE Trans Fuzzy Syst* 2014;22:437-50. DOI
139. He W, Huang H, Ge SS. Adaptive neural network control of a robotic manipulator with time-varying output constraints. *IEEE Trans Cybern* 2017;47:3136-47. DOI PubMed
140. He W, Ouyang Y, Hong J. Vibration control of a flexible robotic manipulator in the presence of input deadzone. *IEEE Trans Ind Inf* 2017;13:48-59. DOI
141. Zhu G, Ge S, Lee T. Simulation studies of tip tracking control of a single-link flexible robot based on a lumped model. *Robotica* 1999;17:71-8. DOI
142. Sun C, He W, Hong J. Neural network control of a flexible robotic manipulator using the lumped spring-mass model. *IEEE Trans Syst Man Cybern, Syst* 2017;47:1863-74. DOI
143. Bertsekas DP, Tsitsiklis JN. Neuro-dynamic programming. Belmont, MA: Athena Scientific; 1996.
144. Pitts W, Mcculloch WS. How we know universals; the perception of auditory and visual forms. *Bull Math Biophys* 1947;9:127-47.

DOI PubMed

145. Liu R. Multispectral images-based background subtraction using Codebook and deep learning approaches. Available from: https://www.theses.fr/2020UBFCA013.pdf [Last accessed on 8 Mar 2022].
146. Liu W, Wang Z, Liu X, Zeng N, Liu Y, Alsaadi FE. A survey of deep neural network architectures and their applications. *Neurocomputing* 2017;234:11-26. DOI
147. Silver D, Schrittwieser J, Simonyan K, et al. Mastering the game of Go without human knowledge. *Nature* 2017;550:354-9. DOI PubMed
148. Laud AD. Theory and application of reward shaping in reinforcement learning. Available from: https://www.proquest.com/openview/bb29dc3d66eccbe7ab65560dd2c4147f/1?pq-origsite=gscholar&cbl=18750&diss=y [Last accessed on 8 Mar 2022].
149. Kober J, Bagnell JA, Peters J. Reinforcement learning in robotics: a survey. *Int J Rob Res* 2013;32:1238-74. DOI PubMed
150. Digney BL. Nested Q-learning of hierarchical control structures. Proceedings of International Conference on Neural Networks (ICNN'96); 1996 Jun 3-6; Washington, DC, USA. IEEE; 1996. p. 161-6. DOI
151. Schaal S. Learning from demonstration. Proceedings of the 9th International Conference on Neural Information Processing Systems; 1996 Dec; Cambridge, MA, USA. IEEE; 1996. p. 1040-6. DOI
152. Kuan C, Young K. Reinforcement learning and robust control for robot compliance tasks. *J Intell Robot Syst* 1998;23:165-82. DOI
153. Bucak IO, Zohdy MA. Application of reinforcement learning control to a nonlinear dexterous robot. Proceedings of the 38th IEEE Conference on Decision and Control (Cat. No.99CH36304); 1999 Dec 7-10; Phoenix, AZ, USA. IEEE; 1999. p. 5108-13. DOI
154. Bucak IO, Zohdy MA. Reinforcement learning control of nonlinear multi-link system. *Eng Appl Artif Intell* 2001;14:563-75. DOI
155. Althoefer K, Krekelberg B, Husmeier D, Seneviratne L. Reinforcement learning in a rule-based navigator for robotic manipulators. *Neurocomputing* 2001;37:51-70. DOI
156. Gaskett C. Q-learning for robot control. Available from: https://digitalcollections.anu.edu.au/bitstream/1885/47080/5/01front.pdf [Last accessed on 8 Mar 2022].
157. Smart WD, Kaelbling LP. Reinforcement learning for robot control. *Proc SPIE* 2002. DOI
158. Izawa J, Kondo T, Ito K. Biological robot arm motion through reinforcement learning. Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No.02CH37292); 2002 May 11-15; Washington, DC, USA. IEEE; 2002. p. 3398-403. DOI
159. Peters J, Vijayakumar S, Schaal S. Reinforcement learning for humanoid robotics. 3rd IEEE-RAS International Conference on Humanoid Robots; 2003 Sep 29-30; Karlsruhe, Germany. 2003.
160. Bhatnagar S, Sutton RS, Ghavamzadeh M, Lee M. Natural actor-critic algorithms. *Automatica* 2009;45:2471-82. DOI
161. Theodorou E, Peters J, Schaal S. Reinforcement learning for optimal control of arm movements. Poster presented at 37th Annual Meeting of the Society for Neuroscience (Neuroscience 2007); San Diego, CA, USA. 2007. DOI
162. Peters J, Schaal S. Natural actor-critic. *Neurocomputing* 2008;71:1180-90. DOI
163. Atkeson CG, Schaal S. Learning tasks from a single demonstration. Proceedings of International Conference on Robotics and Automation; 1997 Apr 25-25; Albuquerque, NM, USA. IEEE; 1997. p. 1706-12. DOI
164. Hoffmann H, Theodorou E, Schaal S. Behavioral experiments on reinforcement learning in human motor control. Available from: https://www.researchgate.net/publication/325463394 [Last accessed on 8 Mar 2022].
165. Peters J, Schaal S. Learning to control in operational space. *Int J Rob Res* 2008;27:197-212. DOI
166. Buchli J, Theodorou E, Stulp F, Schaal S. Variable impedance control - a reinforcement learning approach. In: Matsuoka Y, Durrant-Whyte H, Neira J, editors. Robotics: Science and Systems VI. Cambridge: MIT Press; 2011. DOI
167. Theodorou E, Buchli J, Schaal S. Reinforcement learning of motor skills in high dimensions: a path integral approach. 2010 IEEE International Conference on Robotics and Automation; 2010 May 3-7; Anchorage, AK, USA. IEEE; 2010. p. 2397-403. DOI
168. Kappen HJ. Path integrals and symmetry breaking for optimal control theory. *J Stat Mech* 2005;2005:P11011. DOI
169. Shah H, Gopal M. Reinforcement learning control of robot manipulators in uncertain environments. 2009 IEEE International Conference on Industrial Technology; 2009 Feb 10-13; Churchill, VIC, Australia. IEEE; 2009. p. 1-6. DOI
170. Kim B, Kang B, Park S, Kang S. Learning robot stiffness for contact tasks using the natural actor-critic. 2008 IEEE International Conference on Robotics and Automation; 2008 May 19-23; Pasadena, CA, USA. IEEE; 2008. p. 3832-7. DOI
171. Kim B, Park J, Park S, Kang S. Impedance learning for robotic contact tasks using natural actor-critic algorithm. *IEEE Trans Syst Man Cybern B Cybern* 2010;40:433-43. DOI PubMed
172. Adam S, Busoniu L, Babuska R. Experience replay for real-time reinforcement learning control. *IEEE Trans Syst , Man, Cybern C* 2012;42:201-12. DOI
173. Hafner R, Riedmiller M. Reinforcement learning in feedback control: Challenges and benchmarks from technical process control. *Mach Learn* 2011;84:137-69. DOI
174. Krizhevsky A, Sutskever I, Hinton GE. ImageNet classification with deep convolutional neural networks. *Commun ACM* 2017;60:84-90. DOI
175. Levine S, Finn C, Darrell T, Abbeel P. End-to-end training of deep visuomotor policies. Available from: http://arxiv.org/abs/1504.00702 [Last accessed on 8 Mar 2022].
176. Levine S, Wagener N, Abbeel P. Learning contact-rich manipulation skills with guided policy search. Available from: http://arxiv.org/abs/1501.05611 [Last accessed on 8 Mar 2022].

177. Tai L, Zhang J, Liu M, Boedecker J, Burgard W. A survey of deep network solutions for learning control in robotics: from reinforcement to imitation. Available from: http://arxiv.org/abs/1612.07139 [Last accessed on 8 Mar 2022].
178. Vecerik M, Hester T, Scholz J, et al. Leveraging demonstrations for deep reinforcement learning on robotics problems with sparse rewards. Available from: http://arxiv.org/abs/1707.08817 [Last accessed on 8 Mar 2022].
179. Liu R, Nageotte F, Zanne P, de Mathelin M, Dresp-langley B. Deep reinforcement learning for the control of robotic manipulation: a focussed mini-review. *Robotics* 2021;10:22. DOI
180. Andrychowicz M, Wolski F, Ray A, et al. Hindsight experience replay. Available from: https://arxiv.org/abs/1707.01495v3 [Last accessed on 8 Mar 2022].
181. Gupta A, Savarese S, Ganguli S, Fei-Fei L. Embodied intelligence via learning and evolution. *Nat Commun* 2021;12:5721. DOI PubMed PMC
182. Rajeswaran A, Kumar V, Gupta A, et al. Learning complex dexterous manipulation with deep reinforcement learning and demonstrations. Available from: http://arxiv.org/abs/1709.10087 [Last accessed on 8 Mar 2022].
183. Matas J, James S, Davison AJ. Sim-to-real reinforcement learning for deformable object manipulation. Available from: http://arxiv.org/abs/1806.07851 [Last accessed on 8 Mar 2022].

Bah *et al*. *Intell Robot* 2022;2(1):72-88
DOI: 10.20517/ir.2021.16

Intelligence & Robotics

Research Article

Open Access

Check for updates

Facial expression recognition using adapted residual based deep neural network

Ibrahima Bah[1], **Yu Xue**[1,2]

[1]School of Computer and Software, Nanjing University of Information Science and Technology, Nanjing 210044, Jiangsu, China.
[2]Jiangsu Key Laboratory of Data Science and Smart Software, Jingling Institute of Technology, Nanjing 211169, Jiangsu, China.

Correspondence to: Dr. Ibrahima Bah, School of Computer and Software, Nanjing University of Information Science and Technology, No. 219, Ningliu Road, Pukou District, Nanjing 211169, Jiangsu, China. E-mail: 20205220003@nuist.edu.cn; Prof. Yu Xue, School of Computer and Software, Nanjing University of Information Science and Technology, No. 219, Ningliu Road, Pukou District, Nanjing 211169, Jiangsu, China. E-mail: xueyu@nuist.edu.cn;

How to cite this article: Bah I, Xue Y. Facial expression recognition using adapted residual based deep neural network. *Intell Robot* 2022;2(1):xx. http://dx.doi.org/10.20517/ir.2021.16

Received: 6 Dec 2021 **First Decision:** 21 Feb 2022 **Revised:** 24 Feb 2022 **Accepted:** 3 Mar 2022 **Published:** 22 Mar 2022

Academic Editor: Simon X. Yang **Copy Editor:** Xi-Jun Chen **Production Editor:** Xi-Jun Chen

Abstract

Emotion on our face can determine our feelings, mental state and can directly impact our decisions. Humans are subjected to undergo an emotional change in relation to their living environment and or at a present circumstance. These emotions can be anger, disgust, fear, sadness, happiness, surprise or neutral. Due to the intricacy and nuance of facial expressions and their relationship to emotions, accurate facial expression identification remains a difficult undertaking. As a result, we provide an end-to-end system that uses residual blocks to identify emotions and improve accuracy in this research field. After receiving a facial image, the framework returns its emotional state. The accuracy obtained on the test set of FERGIT dataset (an extension of the FER2013 dataset with 49300 images) was 75%. This proves the efficiency of the model in classifying facial emotions as this database poses a bunch of challenges such as imbalanced data, intraclass variance, and occlusion. To ensure the performance of our model, we also tested it on the CK+ database and its output accuracy was 97% on the test set.

Keywords: Facial expression recognition, emotion detection, convolutional neural network, deep residual network

1. INTRODUCTION

Detecting a person's emotions has become increasingly important in recent years. It has attracted interest, in human emotion detection across a variety of areas including but not limited to human-computer[1], education, and medicine. Interpersonal communication is impossible without emotions coming into play. In the daily life of human communication, emotions play a significant role. Human emotional states can be gleaned from spoken (verbal), and nonverbal information is collected by a variety of sensors. According to the 7-38-55 rule[2], verbal communication accounts for only 7% of all communication, whereas nonverbal components of our daily conversation, such as voice tonality and body language, account for 38% and 55%, respectively. Human emotions are exposed via changes on the face, voice intonation as well as body language. Studies have proven that emotions expressed visually are most prominent which are displayed on individual faces. They can be shown in a variety of ways, some of which are visible to the human eye and others that are not.

Emotion is a multidisciplinary area that includes psychology, computer science, and other disciplines. It can be described in psychology terms as a psychological state that is associated with thoughts, feelings, behavioral reactions, and a level of pleasure or dissatisfaction[3]. Whereas in the field of computer science, it may be recognized in the form of image, audio, video, and text documents. Emotion analysis from any of those document types is not easy. People communicate mostly through their emotional reactions which can be positive, negative, or neutral. It is generally accepted that good emotions are conveyed as a variety of different adjectives such as cheerful, happy, joy, excited, while negative emotions can be hate, anger, fear, depression, sadness and so on. People spend the majority of their time posting and expressing their feelings on social media sites such as Facebook, Instagram, and others[4]. They allow people to express their emotions in many different ways.

In our daily lives, we are faced with situations that affect our emotions. It has a significant impact on human cognitive functions such as perception, attention, learning, memory, reasoning, and problem-solving[5]. Among these, attention is the most impacted, both in terms of altering attention's selectivity and in terms of driving actions and behaviors. Human emotion can have a great impact on their health if poorly managed. It weakens the immune system making it more susceptible to colds and other illnesses[6].

Deep learning's growth has greatly improved the accuracy of facial expression identification tasks. Various Convolutional Neural Network (CNN) models have recently been built to overcome problems with emotion recognition from facial expressions. It is one of the leading networks in this field. A CNN architecture is composed of convolutions, activations, and pooling layers. With the advancement of Artificial Intelligence technologies such as pattern recognition and computer vision, computing terminal devices can now interpret the changes in human expressions to a degree, allowing for greater diversity in human-computer communication[7]. In Facial Expression Recognition (FER), the major aim is to map distinct facial expressions to their corresponding emotional states. It consists of extracting the features from the facial image and recognizing the emotion presented. Before feeding facial images to a CNN or other different machine learning classifier, some image processing techniques need to be done. Existing methods include discrete wavelet transform[8], linear discriminant analysis[9], histogram equalization[10], histogram of gradients[11], viola-jones algorithm[12], *etc*. When it comes to real conditions like occlusion and light, manual feature extraction has a good identification capacity in specific special situations or laboratory environments, but it struggles when it comes to natural conditions. Feature extraction approaches based on deep convolution neural networks have attracted a lot of attention recently[13], and this has helped to improve facial emotion detection performance. Deep Residual Network[14] (Deep ResNet) which was easier to train and optimize, has played a major role in the field of image recognition, introducing a novel approach to Deep Neural Network optimization.

Previous work on emotion recognition depended on a two-stage classical learning strategy. The first stage consists of extracting features using image processing techniques. The second stage, on the other hand, relied on the employment of a traditional machine learning classifier such as Support Vector Machine (SVM) to detect emotions. FER has used a variety of methodologies to extract the visual highlights of picture layouts such as weighted random forest (WRF)[15]. Hasani and Mahoor[16] utilized a novel network called ResNet-LSTM to capture Spatio-temporal data, which combine lower highlights to LSTMs specifically. The deep learning network has ended up as the most widely utilized strategy in FER due to its powerful feature extraction capacity.

Using histogram of oriented gradients (HOG) in the wavelet domain, Nigam *et al.*[11] proposed a four steps process for efficient FER (face processing, domain transformation, feature extraction and expression recognition). In the expression recognition part, the authors used a tree-based multi-class SVM to classify the retrieved HOG features in discrete wavelet transform (DWT). The system was trained and tested with CK+, JAFFE and Yale datasets. The accuracy observed in the test set of these three (3) datasets are 90%, 71.43% and 75% respectively.

Upon deeply analyzing the Facial Expression Recognition problem, Minaee et al. proposed the use of Attentional Convolutional Neural Network[17] instead of adding layers/neurons. Aside from that, they also suggested adding a visualization technique that can find important parts of the face that is necessary for detecting different emotions based on the classifier's output. Their architecture includes a feature extraction part and spatial transformer network that takes the input and uses the affine transformation to wrap it to the output. They achieved a validation accuracy of 70.02 per cent for the categorization of the 7 classes using the FER2013 dataset.

With the help of the Residual Masking Network[18], the authors focused on deep architecture with the attention mechanism. They used a segmentation network to refine feature maps, by enabling the network to focus on relevant information to make the correct decision. Their work was divided into 2 parts: the residual masking block which contains a residual layer, and the ensemble method for the combination with 7 different CNNs. In the end, they managed to get an overall accuracy of 74.14% on the test set of FER2013 dataset.

Pu and Zhu[19] developed a FER framework based on the combination of a feature extraction network and pre-trained model. The feature extraction consists of supervised learning optical flow based on residual block. The classifier is the Inception architecture. By experimenting with their method on CK+ and FER2013 datasets they achieved the average accuracy of 95.74% and 73.11% respectively. In order to resolve the fact that CNNs require a lot of computation resources to train and process emotional recognition, Chowanda[20] proposed a separable CNN. In the experiment, a comparison of four networks has been made. Networks with and without separable modules, using flatten and fully connected layers, and using global average pooling. Their proposed architecture was faster, with fewer parameters and achieved an accuracy of 99.4% on the CK+ dataset.

Deep learning methods have recently sparked a lot of interest, and there is a lot of research going on using deep learning methods to recognize emotions from facial expressions. However, this study proposes the accurate identification of facial emotion using a deep residual-based neural network architecture model. ResNet was chosen as the study's foundation because residual-based network models have shown to be effective in a variety of image recognition applications and have also overcome the problem of overfitting. In our work, we used emotional expressions such as happiness, surprise, anger, sadness, disgust, neutral, and fear to pick up emotional changes on individual faces. Furthermore, the main contribution of this work are:

1. Propose a lighter version of CNN using Residual Blocks with fewer number of trainable parameters compared to over 23 millions for the original ResNet network.
2. Locate the best position to use the Residual Blocks to avoid overfitting, and finally get a satisfying performance.
3. Show the important of using Residual Blocks compared to the architecture without them.
4. Weight the cross-entropy loss function in order to deal with imbalance problem that suffer the FERGIT

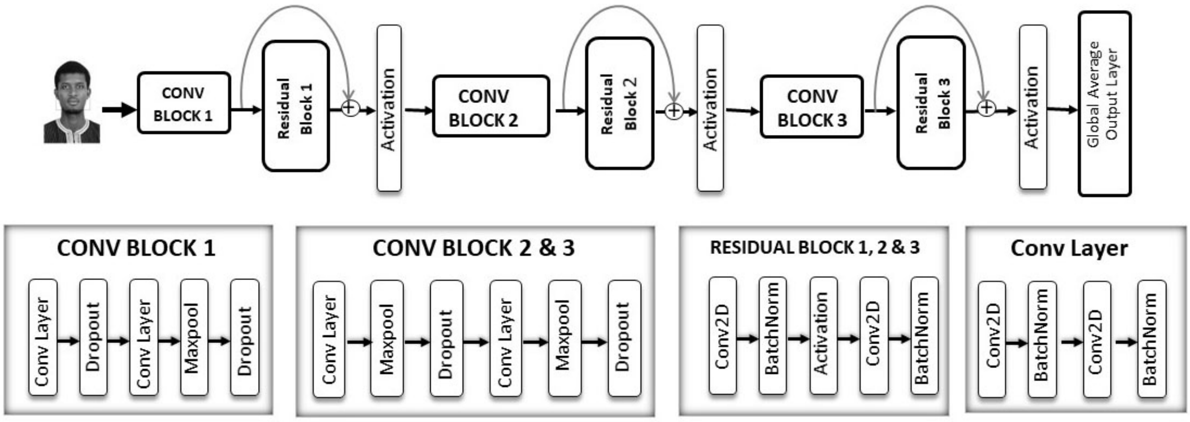


Figure 1. Architecture model of the proposed framework.

dataset.

5. Confirm the validity of the model by the number of parameters, run-time, and accuracy recorded on the Cohn–Kanade (CK+) and FERGIT datasets.

2. METHODS

In this section, we introduce our improved Convolution Neural Network with Residual Blocks. This model is an end-to-end deep learning framework to classify emotions on human face. The model has a total of 7 blocks (3 Convolutional Blocks, 3 Residual Blocks and one Classification Block) in number. This study looked at strategies that can be used indefinitely, such as CNN for quick and responsive systems with short processing and reaction times.

2.1. Proposed model architecture

In our proposed architecture, the feature extraction part consists of twelve convolutional sub-blocks with a Rectified Linear Unit (ReLU) activation function and a kernel initializer set to he_normal in the convolutional layers. A Residual Block is added after every four convolutional layers. This block also called skip connection or identity mapping consists of two convolutional layers, each one followed by a batch normalization layer, and the results from all Residual Blocks are added to the previous convolution and activated. In the basic network, each pair of a layer is followed by a batch normalization layer, max-pooling layer, and dropout layer. They are then followed by a global average pooling layer and a dense layer as the output. In the final output layer, we used the softmax activation function to perform the task of classifying the seven emotions. All details expressed above can be observed in our proposed framework architecture below shown in Figure 1.

In our framework, we located the best positions to use the residual blocks by trial and error means. Thus, the number of parameters has been reduced considerably compared to the original Deep ResNet[14], and the network was fast to train, see Table 1.

2.1.1. Convolution

CNN because of its structure, is arguably the best suitable architecture to use when dealing with computer vision tasks[21]. The basic operation is the convolution operation, it consists of merging two sets of information. The convolutional layer's job is to multiply the previous layer's image pixels by a learnable convolutional kernel at the corresponding place. And then, calculate the weighted sum of the multiplied results[22]. For the first convolution operation, we applied a convolution filter (kernel) of size 5×5 with a stride of 1 and padding of 2, the latter is used to maintain the same shape of the input image. The output shape is obtained using this

Table 1. Architecture detail

Layer (type)	Output shape	Param #
input (InputLayer)	[(None, 48, 48, 1)]	0
conv2d_1 (Conv2D)	(None, 48, 48, 16)	416
batch_normalization_1 (BatchNormalization)	(None, 48, 48, 16)	64
conv2d_2 (Conv2D)	(None, 48, 48, 16)	6416
batch_normalization_2 (BatchNormalization)	(None, 48, 48, 16)	64
dropout_1 (Dropout)	(None, 48, 48, 16)	0
conv2d_3 (Conv2D)	(None, 48, 48, 32)	4640
batch_normalization_3 (BatchNormalization)	(None, 48, 48, 32)	128
conv2d_4 (Conv2D)	(None, 48, 48, 32)	9248
batch_normalization_4 (BatchNormalization)	(None, 48, 48, 32)	128
max_pooling2d_1 (MaxPooling2D)	(None, 24, 24, 32)	0
dropout_2 (Dropout)	(None, 24, 24, 32)	0
conv2d_5 (Conv2D)	(None, 24, 24, 32)	9248
batch_normalization_5 (BatchNormalization)	(None, 24, 24, 32)	128
activation_1 (Activation)	(None, 24, 24, 32)	0
conv2d_6 (Conv2D)	(None, 24, 24, 32)	9248
batch_normalization_6 (BatchNormalization)	(None, 24, 24, 32)	128
add_1 (Add)	(None, 24, 24, 32)	0
activation_2 (Activation)	(None, 24, 24, 32)	0
conv2d_7 (Conv2D)	(None, 24, 24, 64)	18496
batch_normalization_7 (BatchNormalization)	(None, 24, 24, 64)	256
conv2d_8 (Conv2D)	(None, 24, 24, 64)	36928
batch_normalization_8 (BatchNormalization)	(None, 24, 24, 64)	256
max_pooling2d_2 (MaxPooling2D)	(None, 12, 12, 64)	0
dropout_3 (Dropout)	(None, 12, 12, 64)	0
conv2d_9 (Conv2D)	(None, 12, 12, 128)	73856
batch_normalization_9 (BatchNormalization)	(None, 12, 12, 128)	512
conv2d_10 (Conv2D)	(None, 12, 12, 128)	147584
batch_normalization_10 (BatchNormalization)	(None, 12, 12, 128)	512
max_pooling2d_3 (MaxPooling2D)	(None, 6, 6, 128)	0
dropout_4 (Dropout)	(None, 6, 6, 128)	0
conv2d_11 (Conv2D)	(None, 6, 6, 128)	147584
batch_normalization_11 (BatchNormalization)	(None, 6, 6, 128)	512
activation_3 (Activation)	(None, 6, 6, 128)	0
conv2d_12 (Conv2D)	(None, 6, 6, 128)	147584
batch_normalization_12 (BatchNormalization)	(None, 6, 6, 128)	512
add_2 (Add)	(None, 6, 6, 128)	0
activation_4 (Activation)	(None, 6, 6, 128)	0
conv2d_13 (Conv2D)	(None, 6, 6, 256)	295168
batch_normalization_13 (BatchNormalization)	(None, 6, 6, 256)	1024
conv2d_14 (Conv2D)	(None, 6, 6, 256)	590080
batch_normalization_14 (BatchNormalization)	(None, 6, 6, 256)	1024
max_pooling2d_4 (MaxPooling2D)	(None, 3, 3, 256)	0
dropout_5 (Dropout)	(None, 3, 3, 256)	0
conv2d_15 (Conv2D)	(None, 3, 3, 512)	1180160
batch_normalization_15 (BatchNormalization)	(None, 3, 3, 512)	2048
conv2d_16 (Conv2D)	(None, 3, 3, 512)	2359808
batch_normalization_16 (BatchNomalization)	(None, 3, 3, 512)	2048
max_pooling2d_5 (MaxPooling2D)	(None, 1, 1, 512)	0
dropout_6 (Dropout)	(None, 1, 1, 512)	0
conv2d_17 (Conv2D)	(None, 1, 1, 512)	2359808
batch_normalization_17 (BatchNormalization)	(None, 1, 1, 512)	2048
activation_5 (Activation)	(None, 1, 1, 512)	0
conv2d_18 (Conv2D)	(None, 1, 1, 512)	2359808
batch_normalization_18 (BatchNormalization)	(None, 1, 1, 512)	2048
add_3 (Add)	(None, 1, 1, 512)	0
activation_6 (Activation)	(None, 1, 1, 512)	0
global_average_pooling2d_1 (GlobalAveragePooling2D)	(None, 512)	0
dense_1 (Dense)	(None, 7)	3591
activation_7 (Activation)	(None, 7)	0
Total params: 9,773,111 Trainable params: 9,766,391 Non-trainable params: 6,720		

formula:

$$OS = \frac{IS - KS + 2P}{S} + 1 \tag{1}$$

Where IS represents the height, or width, assuming that height = width in this study; KS the shape one of the kernel; P is the padding (here a zero padding is applied) and S represents the stride.

The input grayscale image of size 48×48 going through the first convolution layer will get the same output shape of size 48×48 see details Equation (2).

$$OS = \frac{48 - 5 + 2 \times 2}{1} + 1 = \frac{47}{1} + 1 = 47 + 1 = 48 \tag{2}$$

We started the convolution with 16 filters and increased it by 2 at each block with the final convolutional layer having a filters size of 512. When convolving the first layer to output the feature map, the 5×5 kernel was chosen to achieve a more detailed extraction of face expression of various scales, and also significantly reduce the number of parameters. The more we move deeper, the more the convolution kernels get bigger, the stronger the network learns feature, and the higher is the recognition accuracy. In this work, we have moderately chosen an appropriate number of filters after several trials that led to the reduction of the number of parameters, thus reducing the computational time, and overfitting. The reason for not using that many filters are because, in FER, the main parts where the networks should focus on are the mouth, towards the corners of the lips, the nose, the eyebrows, the crow's feet, the eyelids, and the eyes[23].

2.1.2. *Rectified linear unit*

The convolution operation given by the following formula:

$$\sum x \times k + b \tag{3}$$

Where x is the input, k the weight and b the bias. Equation (3) is linear, so it follows the mathematical rules:

$$f(x + y) = f(x) + f(y) \tag{4}$$

$$f(\alpha x) = \alpha x \tag{5}$$

Therefore, to avoid the entire network from collapsing into a single equivalent convolutional layer, the use of a nonlinear activation function is needed[24]. Rectified Linear Unit (ReLU)[25], is one of the most used nonlinear activation functions for convolution layers from studied literature[25]. Its function is :

$$f(x) = max(0, x) \tag{6}$$

Where x is the input, and the result will be 0 if $x < 0$ and x if $x > 0$. We used this activation function in our study as we realized that the framework being deep, it reduces considerably the training time.

2.1.3. *Initializers*

Bias in the neural network is like a constant in a linear function, and research has proved that it plays an important role in a Convolutional Neural Network. It helps the model to match the given data better by adjusting the output[26]. The goal of initializing the weights and bias is to keep layer activation outputs from bursting or disappearing during a deep neural network forward pass[27], because if it does happen, the gradients will be either too large or too tiny, causing the network to converge slowly or to not converge at all. He Normal[28] weight initialization has been used in this study. In this case, the weights are randomly initialized and multiplied by the following formula:

$$\sqrt{\frac{2}{size\_l - 1}} \tag{7}$$

Where $size\_l - 1$ is the size of the layer $l - 1$. This strategy ensures that the weights are neither too large nor too small. The biases are initialized to zero since it's the common technique and it proved to be efficient.

2.1.4. Batch-normalization

Batch-Normalization (BN) is a regularization technique[29] that speeds up and stabilizes the training of Deep Neural Networks (DNN). BN avoids the problem of massive gradient updates, which cause divergent loss and uncontrollable activation as network depth increases. As a result, it entails using the current batch's mean and variance to normalize activation vectors from hidden layers[30]. In this research, we placed the BN layer after the activation in the simple Convolutional Blocks and before the activation in the Residual blocks, see Figure 1.

2.1.5. Max pooling

Pooling is performed to reduce the dimensionality of the convolved image[24]. By applying pooling operation, we reduce the number of parameters and fight against overfitting. Max pooling concerns taking the maximum pixels in the size of the given windows[31]. During this process, the model does not learn. In our work, we took a 2×2 window size and strides of 2 for the whole max-pooling layers. The output size is also given by the Equation (1), where padding is 0. Using these parameters, we divide the height and width of each feature map by 2.

2.1.6. Dropout

Dropout[32] is by far the most used Deep Neural Network regularization approach. It boosts the accuracy of the model and avoids overfitting. The idea of using dropout is to randomly prevent some neurons at one step to fire with a frequency of rate p[33], while the other neurons are scaled up by $1/(1-p)$ so that the sum inside the neuron remains unchanged. The same neuron can be actif at the next step and so on so forth. p is the hyper-parameter of the dropout layer, in our study we found out that the best value of p is 0.3 for the early layers of the feature extraction part and 0.4 for the last Convolutional Block.

2.1.7. Residual block

The Residual Block also known as identity shortcut connection was used in our study. It has a function of

$$H(x) = R(x) + x \tag{8}$$

Where $H(x)$ represents the output learning, x is the input and $R(x)$ is the residual layer[14]. The advantage of this network in our study is that it reduced considerably the loss during the training and increased the accuracy on the test set. The residual block is used to solve the problem of vanishing gradients. By skipping some connections, we will allow the back-propagation towards the entire network and so give better performance. In our implementation, we discovered that using the shortcut branch of 1×1 convolution is not suitable as it does not help to reduce the overfitting, see Figure 1.

2.1.8. Global average pooling

Most of the research in CNN use flatten layer[34] to wrap up into a 1D vector the extracted features from previous convolutional layers and forward them to the fully connected layers. Global Average Pooling is a pooling technique used to substitute fully connected layers in traditional CNNs[22]. In this study using the average pooling layer, the resulting vector, the average of each feature map is fed directly into the softmax layer instead of constructing fully connected layers on top of the feature maps.

2.2. Data description

In this study, we mainly used the FERGIT dataset which is a combination of the FER-2013 and muxspace datasets. The FER2013 database was collected from the internet, and most pictures were captured in the wild using search engine research. It appears to be a low human FER system with an accuracy of about 65%[35]. The FERGIT dataset comprises 49,300 detected faces in a grayscale of 48-by-48 pixels. The images shown in Figure 2 are sample emotions from the FER2013 dataset.

The FER2013 has many problems itself, thus making it very difficult for deep learning architecture to achieve

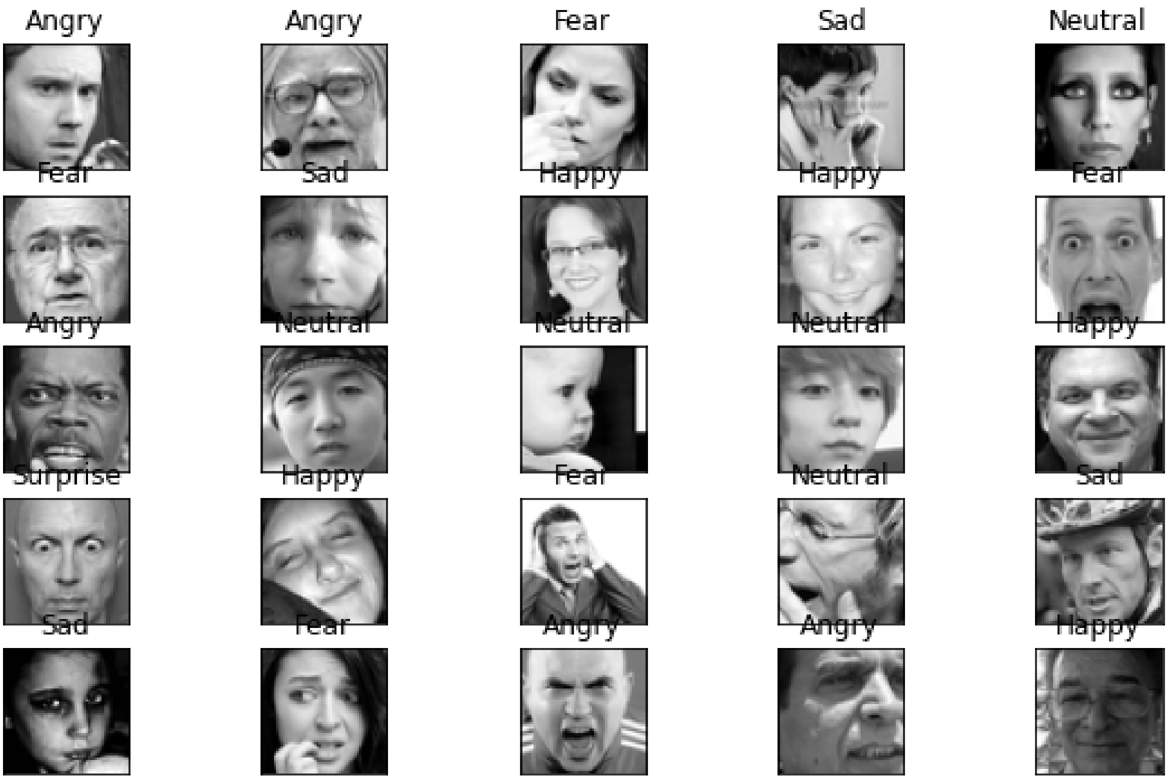


Figure 2. The seven expressions included in the FER-2013 dataset (anger, fear, happiness, sadness, surprise, disgust, and neutral).

better results with its data. Some major issues are imbalanced data, intra-class variation, and occlusion. The FERGIT database is a largely imbalanced dataset, in the training data, classes have huge different number of samples. the happy emotion has more than 13 thousand samples, whereas the disgust has just six hundred samples see Figure 3.

The intra-class variation is the variance within the same class. Minimizing intra-class variation whiles maximizing inter-class variation has a significant effect on classification. Variations, uncontrolled illusions, and occlusions are problems that face recognition systems face in real-life applications[36]. These problems lead to accuracy degradation compared to dataset experimental test performance. A facial occlusion posture is one of several potential stances in which something blocks (occludes) a portion of a person's face, such as their hand. Occlusion might be caused by one or both hands being immediately on or in front of the face. Likewise, hair, caps, and sunglasses are all common items that obstruct the view of the face. Despite occlusion posing a challenge to face recognition, they could potentially yield valuable information because people face using their hands when communicating via gestures.

2.3. Data preprocessing

First, we arbitrarily partitioned the training information into three parts: 44,370 faces (90% of the dataset) were used for preparing our model, 2465 (5% of the dataset) faces for validation, and 2465 (5% of the dataset) faces for testing, as detailed in Figure 4. The size of the dataset is relatively small; therefore, there is a need to augment the dataset to create new data that the model has not seen before from the training set.

Since the FERGIT dataset only contains facial images[35], no face detection, localization, cropping, or face alignment were performed during data preparation for the training. We only performed those steps when

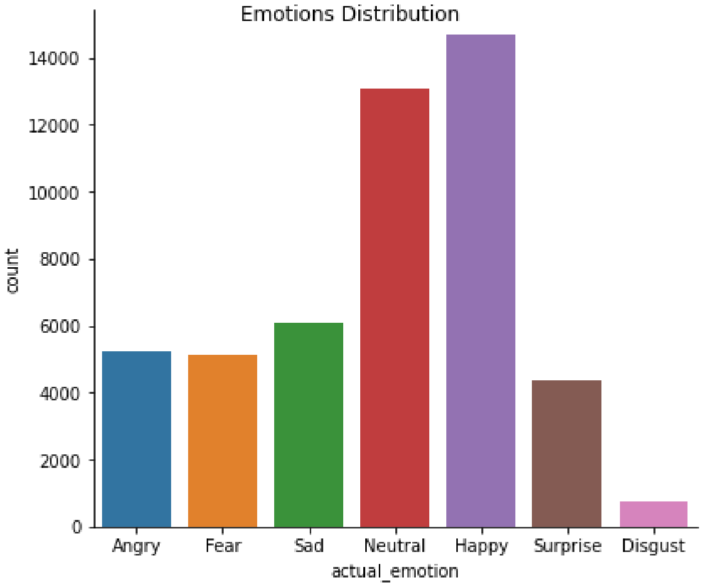


Figure 3. Data distribution among the 7 emotions.

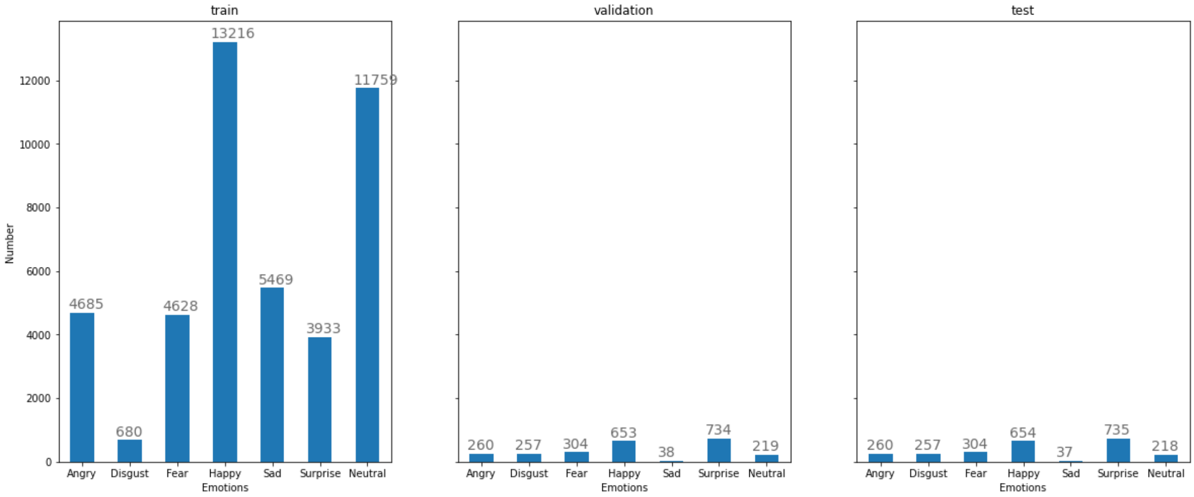


Figure 4. Data split into training, validation and testing sets.

testing with a personal image. Data augmentation improves and enhances training dataset's image size and quality via suitable techniques[37]. The problem of overfitting that is common from the lack of sufficient data is reduced through data augmentation[38]. This research uses data augmentation to transform an image to its original state and train the CNN architecture.

The data augmentation is done by applying the geometrical transformation by first creating a new set of the horizontally flipped datasets, image rotation, shift, and zoom, among other transformation operations[37], and adjusting the brightness to create new images of the same face. The images are also normalized to make the pixel values range from 0 to 1. The provided images are then ready to be used to train the model.

2.4. Training phase

In this work, the models were trained on google colab pro with GPU availability, and they were implemented using Keras2 and Python3. The training was conducted in two phases. We trained the network with a deep Convolutional Neural Network without Residual Blocks in the first phase. Then after noticing that the accuracy is not increasing that much, we added a residual block to help the network generalize well and improve the success rate. The two models were allowed to train for 100 epochs with a batch size of 64. The optimizer used to train the models is the Nadam[39] based on the stochastic gradient descent algorithm, with a learning rate of 0.001, beta_1 parameter value of 0.9, beta_2 is 0.999. and the loss function used is the categorical cross-entropy[40] function since we have a model with more than two outputs. For this work, having a problem of imbalance data, we highly weighted the classes with few number of samples and gave small weights to those with big number of samples. The learning rate is regulated during the training by the callback class ReduceLROnPlateau[41] implemented in the Keras library. This class has the particularity to update the learning to the minimum value (min_lr = 0.0000001) when there is no improvement of the validation accuracy and will stop the training after 15 epochs. We chose 15 epochs to allow the training to last for a long time. Another callback class used is the EarlyStopping[42]. The patience here is set to 30, and finally, we used the ModelCheckpoint to save the model after each improvement of the validation accuracy.

3. RESULTS

The training process of the two models respectively the basic CNN and ResNet based CNN on FERGIT dataset, and the ResNet based CNN on CK+ dataset took only 119 minutes of total training time with colab pro (K80 GPUs, 25GB RAM).

3.1. Performance Analysis

To efficiently evaluate the performance of our model, several metrics have been taken into account. They are precision, recall, F1-Score and accuracy. The recall also called sensitivity is the true positive rate. The precision is to give details about what is the proportion of the correctly predicted positive. The balance of these two metrics is given by the f1-score metrics. Accuracy, the most used metric for classification tasks, is used to find what is the correctly predicted positive and negative in the total test set. Details of the equations are given below:

$$Recall = \frac{TP}{TP + FN} \tag{9}$$

$$Precision = \frac{TP}{TP + FP} \tag{10}$$

$$F1 - Score = 2 \times \frac{Recall \times Precision}{Recall + Precision} \tag{11}$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{12}$$

Where (TP) represents True Positives or where predictions for each emotion were accurately identified. (TN) represents True Negatives or where the model properly rejected a class prediction. (FP) represents False Positives or where predictions for a certain class were wrongly recognized. (FN) represents False Negatives or where the model erroneously rejected for a certain class. The confusion matrix is an important tool for efficiency estimation as it gives a direct comparison of the real and predicted labels.

The first attempt using the basic Deep Neural Network gave an accuracy of 75% on the training data and 73.7% on the validation data. There was no overfitting of the model as, after each convolutional layer, batch normalization is added to ensure that the weights are re-centered. But we realized that even after training the model for more epochs, lasting for only 44 minutes, the maximum accuracy was 75%, and the model gave a 74% success rate on the test set, as mentioned in Table 2.

Table 2. Basic model classification performance test results on FERGIT

	Precision	Recall	F1-Score	Support
Angry	0.64	0.49	0.56	260
Disgust	0.75	0.57	0.65	37
Fear	0.6	0.44	0.51	257
Happy	0.89	0.93	0.91	735
Sad	0.54	0.62	0.57	304
Surprise	0.76	0.73	0.75	218
Neutral	0.75	0.82	0.78	654
Accuracy			0.74	2465
Run time			44 min	

Table 3. ResNet based model classification performance test results on FERGIT

	Precision	Recall	F1-Score	Support
Angry	0.62	0.54	0.58	260
Disgust	0.71	0.54	0.62	37
Fear	0.62	0.44	0.51	257
Happy	0.89	0.92	0.9	735
Sad	0.55	0.56	0.55	304
Surprise	0.76	0.79	0.77	218
Neutral	0.75	0.84	0.79	654
Accuracy			0.75	2465
Run time			48 min	

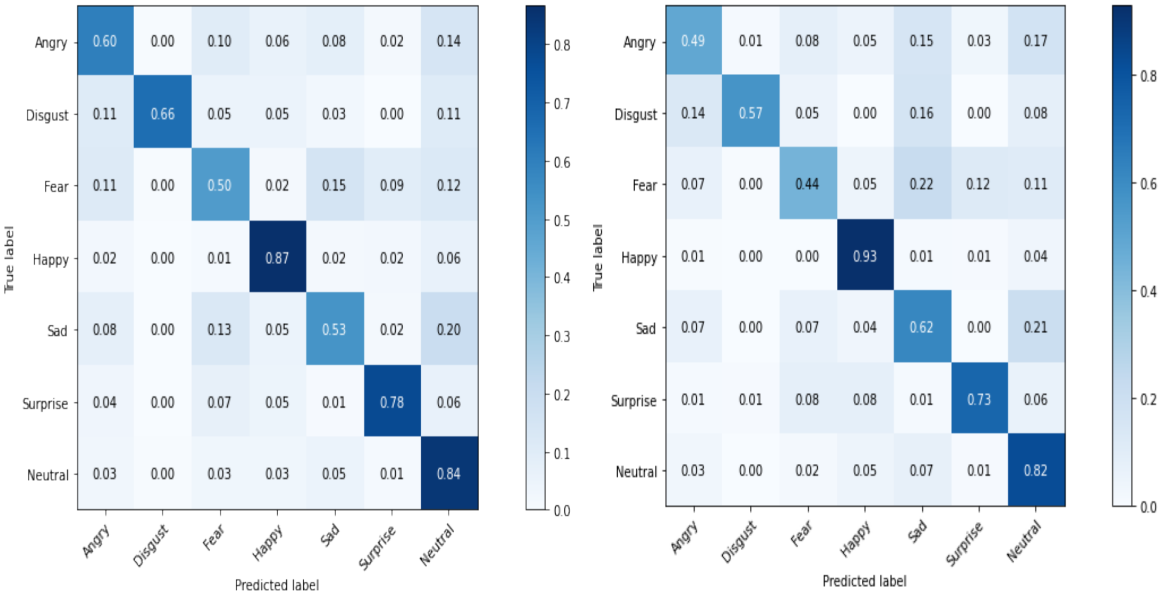
To improve the success rate and at the same time reduce the loss, we used residual blocks, which proved to be efficient as the accuracy increased to 86% on the training data, and we finally got an accuracy of 75% on the test set as shown in Table 3. This training took more time, running for 48 minutes.

The model does well on disgust, happiness, surprise, and neutral or contempt expressions during the two phases. Despite the very imbalanced training data that is alleviated with class-weighting loss -the happy label has around 30% of the test split- our model's overall performance was quite good, as presented in the confusion matrix (See Figure 5). It can be seen that the residual-based network balanced the performance versus the basic network that biased more on the neutral and happy classes. In both cases, 93% of the images labelled happy were truly predicted while the prediction of fear was not good, barely 50% for the residual-based model. This is due to the mislabeling of most of the images.

3.2. Accuracy and loss during training

For the first attempt with the basic network, we observe that the model is learning very well in the training data and generalizing to the validation data of the FERGIT database. There was no overfitting during 100 training epochs, but the overall accuracy did not increase much, and the network did not stop the training. We increased the number of epochs to seek better performance, but the network stuck to 75%, the best the model could achieve. And which evaluated to the test set it achieved a perfect accuracy of 74%. The loss rate was 0.48% on the train set, 0.79% on the validation set, and 0.7% on the test set. See Figure 6.

In the second experiment, the accuracy increased a little bit with the help of the residual blocks. Using them allowed the model to propagate to the early layers and adjust all the weights to get a better result. We observe that after 35 epochs, the model was not generalizing well anymore. Nonetheless, we did not discontinue training because the training loss rapidly decreased while the validation loss was stable. However, the accuracy, on the other hand, was increasing. The model achieved a training accuracy of 86%, validation accuracy of 74%, and test accuracy of 75%. And the loss rate was 0.2% on the train set, 0.82% on the validation set and 0.8% on the test set, See Figure 7.



	Angry	Disgust	Fear	Happy	Sad	Surprise	Neutral
Angry	0.60	0.00	0.10	0.06	0.08	0.02	0.14
Disgust	0.11	0.66	0.05	0.05	0.03	0.00	0.11
Fear	0.11	0.00	0.50	0.02	0.15	0.09	0.12
Happy	0.02	0.00	0.01	0.87	0.02	0.02	0.06
Sad	0.08	0.00	0.13	0.05	0.53	0.02	0.20
Surprise	0.04	0.00	0.07	0.05	0.01	0.78	0.06
Neutral	0.03	0.00	0.03	0.03	0.05	0.01	0.84

	Angry	Disgust	Fear	Happy	Sad	Surprise	Neutral
Angry	0.49	0.01	0.08	0.05	0.15	0.03	0.17
Disgust	0.14	0.57	0.05	0.00	0.16	0.00	0.08
Fear	0.07	0.00	0.44	0.05	0.22	0.12	0.11
Happy	0.01	0.00	0.00	0.93	0.01	0.01	0.04
Sad	0.07	0.00	0.07	0.04	0.62	0.00	0.21
Surprise	0.01	0.01	0.08	0.08	0.01	0.73	0.06
Neutral	0.03	0.00	0.02	0.05	0.07	0.01	0.82

Figure 5. Confusion matrix for the FER showing the prediction accuracy of the emotional expressions using the proposed architecture with residual blocks (left) and the basic architecture (right).

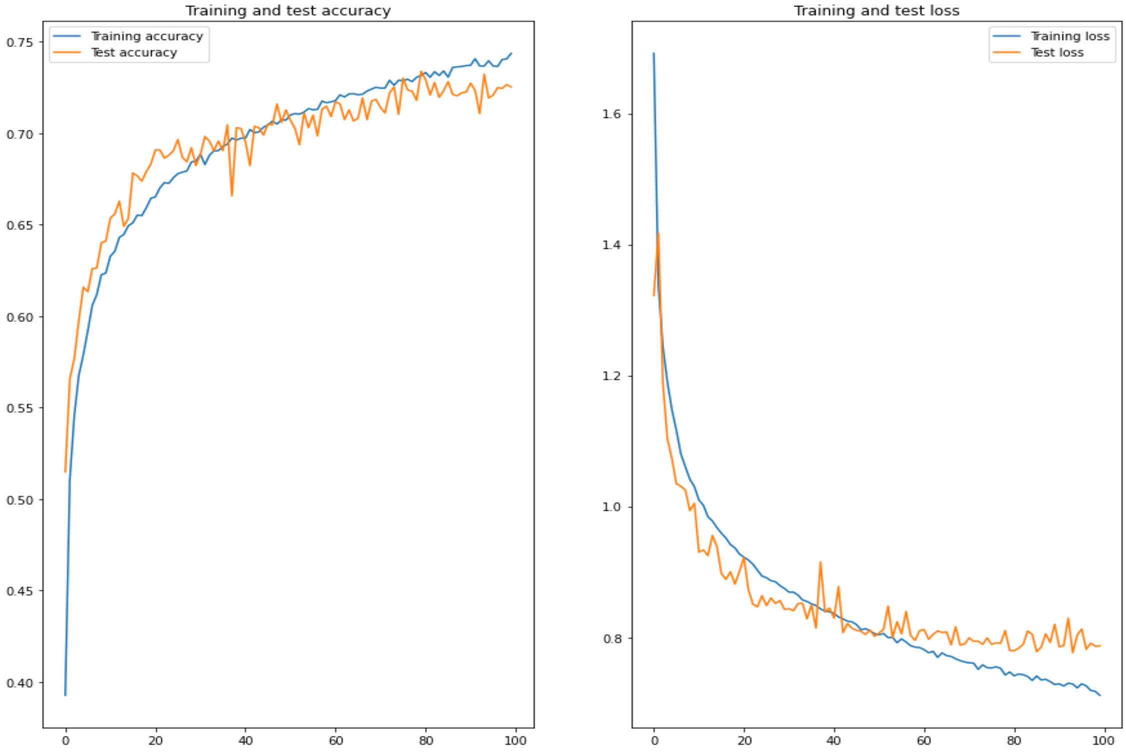


Figure 6. The accuracy and the loss of the training and validation data of FERGIT on the basic model over 100 epochs.

To validate our network's capability of fast generalization and giving the best accuracy, we also trained it on the CK+ database[43]. The CK+ database is relatively small, with 981 samples well partitioned with seven classifications of emotions: Angry, Disgust, Fear, Happy, Sadness, Surprise, and Contempt[43]. Using dropout layers helped the model train on a very small dataset (see Figure 8). The model achieved competitive results on

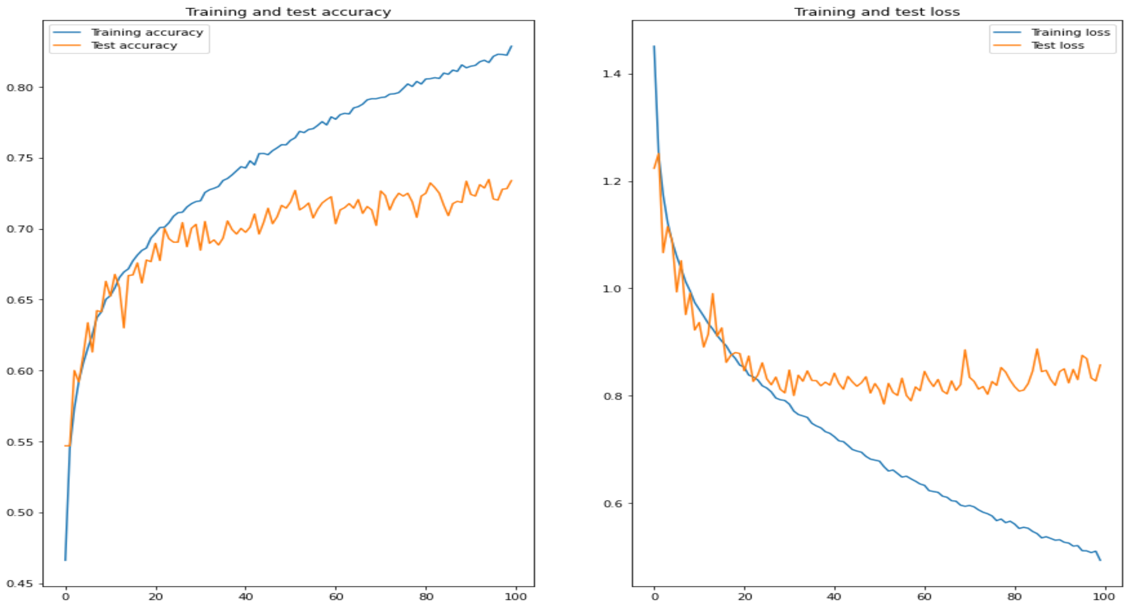


Figure 7. The accuracy and the loss of the training and validation data of FERGIT on the ResNet based model over 100 epochs.

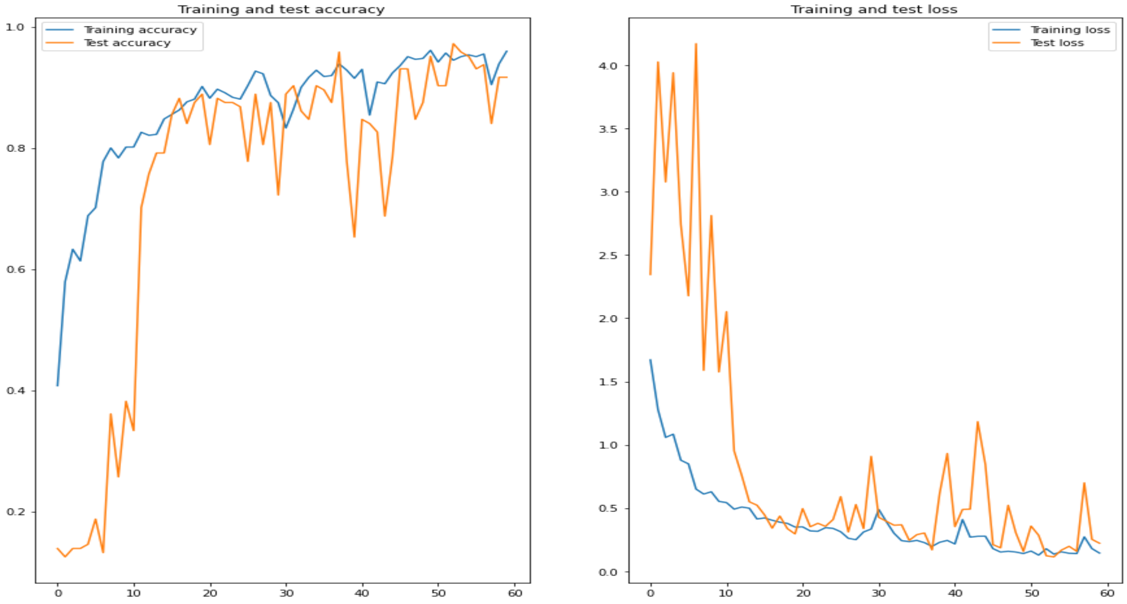


Figure 8. The accuracy and the loss of the training and validation data of CK+ on the basic model over 100 epochs.

CK+, 97%. See Table 4. These results express the superiority of the presented methodology compared to best results with CNN architectures such as Pu[19] who achieved an accuracy of 95.74%, and Cheng[44] achieved success rate of 94.4%. See Table 5.

4. DISCUSSION

Our CNN FER model, which is based on ResNet, took 48 minutes to learn multiple facial images and then distinguish between seven (7) emotions, although the number of parameters is relatively big (9,766,391 parameters). The traditional CNN on the other hand took 44 minutes to run 100 epochs, but without improving

Table 4. ResNet based model classification performance test results on CK+

	Precision	Recall	F1-Score	Support
Anger	1.00	1.00	1.00	14
Contempt	1.00	0.6	0.75	5
Disgust	1.00	0.94	0.97	18
Fear	0.88	1.00	0.93	7
Happy	0.91	1.00	0.95	21
Sadness	1.00	1.00	1.00	9
Surprise	1.00	1.00	1.00	25
Accuracy			0.97	99
Run time			29 min	

Table 5. Comparison with other CNN based models results on CK+

Methodology	Accuracy (%)
Diff ResNet [19]	95.74
Improved VGG-19 [44]	96
Ours	**97**

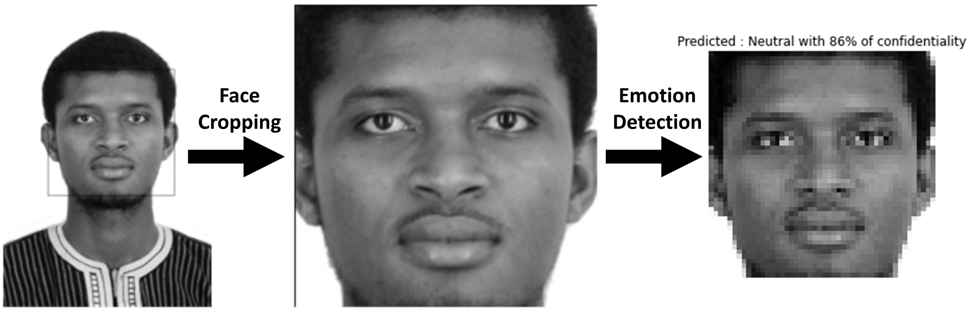


Image Dataset For Model

Facial Crop For Model Testing

Predicted Emotion by Model

Figure 9. Framework testing on an individual pose

much in terms of accuracy. The goal is to build a robust and accurate model. Therefore, looking at Table 2 and Table 3, we observe that for the two models the precisions of all the labels are over 50%, this is to say that for each emotion at least 50% time the model is giving a good prediction. The harmonic mean of these of the recall and precision hereafter referred to as the f1-score, is utilized to determine how well the model performs in terms of facial emotion detection. The value of the f1-score in both cases is 65%, that value is very acceptable regarding how complex the dataset is. Finally, given that the Residual based model has a large 1% plus, accuracy was the metric that helped us identify the optimal model.

We experimented on posed images of an individual to test how well our system recognizes facial expressions that have been previously trained. The first step is to apply some pre-processing such as face detection and face cropping on these images. The image is reshaped to (48,48,1) to have the same shape as the model is trained on, then we use the model for the prediction as shown in Figure 9.

On this prediction, the model did very well with a high percentage of confidentiality (86%), which shows how efficient our framework is. Therefore, the model gave wrong predictions on some labels, but with a low percentage (40%), it's due to the resemblance of emotions, sees Figure 10. For example, here, the individual was asked to pose disgust, but the model predicted neutral. And as already mentioned, the FERGIT dataset has a lot of mis-classified emotions.

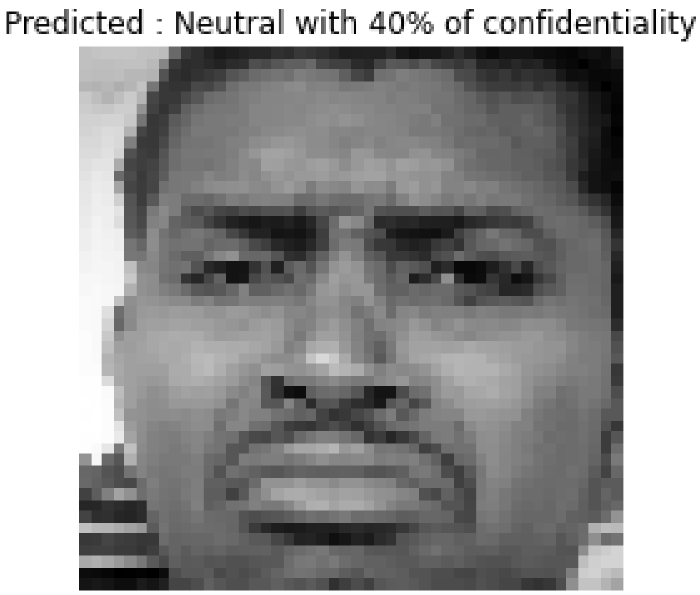


Figure 10. Wrong prediction on the individual facial image.

5. CONCLUSIONS

This paper proposes a novel model of improved CNN architecture with Residual Blocks for Facial Expression Recognition. We evaluated the model on two datasets and compare it to a network without Residual Blocks. The results proved that the proposed architecture performed very well with an accuracy level of 75% on FERGIT challenging dataset. With a relatively big number of parameters (9,766,391), the model achieved a state-of-the-art result in 48 min after running for 100 epochs.This study dataset was augmented to generate similar images so that the model can quickly detect the emotion on the face . Hence, our proposed model shows an overfitting issue during training, affecting the classification. In the future, we look forward to reducing the overfitting and increasing the performance by using more image pre-processing and data enhancement to tackle the occlusion problem. Also, introduce hybrid loss function to handle the intraclass variation problem, and work more on the CNN architecture like using evolutionary computation algorithms to find the best model and optimize the parameters.

DECLARATIONS

Authors' contributions

Made substantial contributions to the conception and design of the study and performed data analysis and interpretation: Bah T, Yu X

Availability of data and materials

The FERGIT dataset is available here: https://www.kaggle.com/uldisvalainis/fergit. The CK+ dataset is available here: https://www.kaggle.com/shawon10/ckplus.

Financial support and sponsorship

None.

Conflicts of interest

All authors declared that there are no conflicts of interest.

Ethical approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

Copyright

REFERENCES

1. Cowie R, Douglas-cowie E, Tsapatsoulis N, et al. Emotion recognition in human-computer interaction. *IEEE Signal Process Mag* 2001;18:32-80. DOI
2. Education, Alice Springs Mparntwe. 5E learning model, 284–5 7–38–55 Rule of Personal Communication, 36. *feedback* 2015;248:50.
3. Parkinson B, Manstead ASR. Current emotion research in social psychology: thinking about emotions and other People. *Emotion Review* 2015;7:371-80. DOI
4. Waterloo SF, Baumgartner SE, Peter J, Valkenburg PM. Norms of online expressions of emotion: Comparing Facebook, Twitter, Instagram, and WhatsApp. *New Media Soc* 2018;20:1813-31. DOI
5. LeBlanc VR, McConnell MM, Monteiro SD. Predictable chaos: a review of the effects of emotions on attention, memory and decision making. *Adv Health Sci Educ Theory Pract* 2015;20:265-82. DOI
6. Luz PM, Brown HE, Struchiner CJ. Disgust as an emotional driver of vaccine attitudes and uptake? A mediation analysis. *Epidemiol Infect* 2019;147:e182. DOI
7. Yonghao Z. Research on the human-computer interaction design in mobile phones. *2020 International Conference on Computing and Data Science (CDS)* IEEE, 2020:395–399.
8. Chervyakov N, Lyakhov P, Kaplun D, Butusov D, Nagornov N. Analysis of the quantization noise in discrete wavelet transform filters for image processing. *Electronics* 2018;7:135. DOI
9. Muslihah I, Muqorobin M. Texture characteristic of local binary pattern on face recognition with probabilistic linear discriminant analysis. *IJCIS* 2020;1:22-6. DOI
10. Pitaloka DA, Wulandari A, Basaruddin T, Liliana DY. Enhancing CNN with preprocessing stage in automatic emotion recognition. *Procedia Computer Science* 2017;116:523-9. DOI
11. Nigam S, Singh R, Misra AK. Efficient facial expression recognition using histogram of oriented gradients in wavelet domain. *Multimed Tools Appl* 2018;77:28725-47. DOI
12. Deshpande NT, Ravishankar S. Face detection and recognition using viola-jones algorithm and fusion of PCA and ANN. *Adv Comput Sci Tech* 2017;10;5:1173–89.
13. Chen Y, Jiang H, Li C, Jia X, Ghamisi P. Deep feature extraction and classification of hyperspectral images based on convolutional neural networks. *IEEE Trans Geosci Remote Sensing* 2016;54:6232-51. DOI
14. He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. In:2016 Proceedings of the IEEE conference on computer vision and pattern recognition. IEEE, 2016, pp. 770–8.
15. Liu ZS, Siu WC, Huang JJ. Image super-resolution via weighted random forest. In:2017 IEEE International Conference on Industrial Technology (ICIT). IEEE 2017, pp. 1019-23.
16. Hasani B, Mahoor MH. Spatio-temporal facial expression recognition using convolutional neural networks and conditional random fields. In:2017 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017). IEEE, 2017, pp. 790-5. DOI
17. Minaee S, Minaei M, Abdolrashidi A. Deep-emotion: facial expression recognition using attentional convolutional network. *Sensors (Basel)* 2021;21:3046. DOI
18. Pham L, Vu TH□ Tran TA. Facial expression recognition using residual masking network. In: 2020 25th International Conference on Pattern Recognition (ICPR). IEEE, 2021, pp. 4513-9. DOI
19. Pu L, Zhu L. Differential residual learning for facial expression recognition. In: 2021 The 5th International Conference on Machine Learning and Soft Computing. IEEE, 2021, pp.103-8. DOI
20. Chowanda A. Separable convolutional neural networks for facial expressions recognition. *J Big Data* 2021;8;1-17. DOI
21. Lee JH, Kim DH, Jeong SN. Diagnosis of cystic lesions using panoramic and cone beam computed tomographic images based on deep learning neural network. *Oral Dis* 2020;26:152-8. DOI
22. Lin M, Chen Q, Yan S. Network in network. arXiv preprint arXiv:1312.4400 2013.
23. Zahara L, Musa P, Wibowo EP, Karim I, Musa SB. The facial emotion recognition (FER-2013) dataset for prediction system of micro-expressions face using the convolutional neural network (CNN) algorithm based raspberry Pi. In: 2020 Fifth International Conference on Informatics and Computing (ICIC). IEEE, 2020, pp. 1-9. DOI
24. Albawi S, Mohammed TA, Al-Zawi S. Understanding of a convolutional neural network. In: 2017 International Conference on Engineering and Technology (ICET). IEEE, 2017, pp. 1-6. DOI
25. Agarap AF. Deep learning using rectified linear units (relu). arXiv preprint arXiv:1312.4400, 2018.

26. Liu Y, Chen Y, Wang J, Niu S, Liu D, Song H. Zero-bias deep neural network for quickest RF signal surveillance. arXiv preprint arXiv:2110.05797, 2021.
27. Hanin B, Rolnick D. How to start training: The effect of initialization and architecture. arXiv preprint arXiv:1803.01719, 2018.
28. Datta L. A survey on activation functions and their relation with xavier and he normal initialization. arXiv preprint arXiv:2004.06632, 2020.
29. Bjorck J, Gomes C, Selman B, Weinberger KQ. Understanding batch normalization. arXiv preprint arXiv:1806.02375, 2018.
30. Santurkar S, Tsipras D, Ilyas A, Mądry A. How does batch normalization help optimization?. In: Proceedings of the 32nd international conference on neural information processing systems. 2018, pp. 2488-98.
31. You H, Yu L, Tian S, et al. MC-Net: Multiple max-pooling integration module and cross multi-scale deconvolution network. *Knowledge-Based Systems* 2021;231:107456. DOI
32. Hinton GE, Srivastava N, Krizhevsky A, Sutskever I, Salakhutdinov R. Improving neural networks by preventing co-adaptation of feature detectors. *CoRR* 2012;abs/1207.0580. Available from http://arxiv.org/abs/1207.0580
33. Yarin G, Jiri H, Alex K. Concrete dropout. arXiv preprint arXiv:1705.07832, 2017.
34. Chen H, Chen A, Xu L, et al. A deep learning CNN architecture applied in smart near-infrared analysis of water pollution for agricultural irrigation resources. *Agricultural Water Management* 2020;240:106303. DOI
35. Goodfellow IJ, Erhan D, Luc Carrier P, et al. Challenges in representation learning: a report on three machine learning contests. *Neural Netw* 2015;64:59-63. DOI
36. Song L, Gong D, Li Z, Liu C, Liu W. Occlusion robust face recognition based on mask learning with pairwise differential siamese network. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. IEEE, 2019, pp. 773-82.
37. Shorten C, Khoshgoftaar TM. A survey on image data augmentation for deep learning. *J Big Data* 2019;6:1-48. DOI
38. Gao X, Saha R, Prasad MR, et al. Fuzz testing based data augmentation to improve robustness of deep neural networks. In: 2020 IEEE/ACM 42nd International Conference on Software Engineering (ICSE). IEEE, 2020, pp. 1147-58.
39. Halgamuge MN, Daminda E, Nirmalathas A. Best optimizer selection for predicting bushfire occurrences using deep learning. *Nat Hazards* 2020;103:845-60. DOI
40. Zhang Z, Sabuncu MR . Generalized cross entropy loss for training deep neural networks with noisy labels. In: 32nd Conference on Neural Information Processing Systems (NeurIPS). 2018.
41. Han Z. Predict final total mark of students with ANN, RNN and Bi-LSTM. Available from http://users.cecs.anu.edu.au/~Tom.Gedeon/conf/ABCs2020/paper/ABCs2020_paper_v2_135.pdf.
42. Li M, Soltanolkotabi M, Oymak S. Gradient descent with early stopping is provably robust to label noise for overparameterized neural networks. In: International conference on artificial intelligence and statistics. PMLR, 2020, pp. 4313-24.
43. Lucey P, Cohn JF, Kanade T, et al. The extended Cohn-Kanade dataset (CK+): A complete dataset for action unit and emotion-specified expression. In: 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Workshops, CVPRW 2010. IEEE, 2010, pp. 94-101. DOI
44. Cheng S, Zhou G. Facial expression recognition method based on improved VGG convolutional neural network. *Int J Patt Recogn Artif Intell* 2020;34:2056003. DOI

Li *et al*. *Intell Robot* 2022;2(1):89–104
DOI: 10.20517/ir.2022.02

Intelligence & Robotics

Research Article

Open Access

An open-closed-loop iterative learning control for trajectory tracking of a high-speed 4-dof parallel robot

Qiancheng Li, Enyu Liu, Chuangchuang Cui, Guanglei Wu

School of Mechanical Engineering, Dalian University of Technology, Dalian 116024, Liaoning, China.

Correspondence to: Assoc. Prof./Dr. Guanglei Wu, School of Mechanical Engineering, Dalian University of Technology, No.2 Linggong Road, Ganjingzi District, Dalian 116024, Liaoning, China. E-mail: gwu@dlut.edu.cn

How to cite this article: Li Q, Liu E , Cui C, Wu G. An open-closed-loop iterative learning control for trajectory tracking of a high-speed 4-dof parallel robot. *Intell Robot* 2022;2(1):89-104. http://dx.doi.org/10.20517/ir.2022.02

Received: 21 Jan 2022 **First Decision:** 3 Mar 2022 **Revised:** 10 Mar 2022 **Accepted:** 14 Mar 2022 **Published:** 31 Mar 2022

Academic Editors: Simon X. Yang, Tao Ren **Copy Editor:** Jia-Xin Zhang **Production Editor:** Jia-Xin Zhang

Abstract

Precise control is of importance for robots, whereas, due to the presence of modeling errors and uncertainties under the complex working environment, it is difficult to obtain an accurate dynamic model of the robot, leading to decreased control performances. This work presents an open-closed-loop iterative learning control applied to a four-limb parallel Schönflies-motion robot, aiming to improve the tracking accuracy with high movement, in which the controller can learn from the iterative errors to make the robot end-effector approximate to the expected trajectory. The control algorithm is compared with classical D-ILC, which is illustrated along with an industrial trajectory of pick-and-place operation. External repetitive and non-repetitive disturbances are added to verify the robustness of the proposed approach. To verify the overall performance of the proposed control law, multiple trajectories within the workspace, different working frequencies for a prescribed trajectory, and different design methods are selected, which show the effectiveness and the generalization ability of the designed controller.

Keywords: High-speed parallel robot, open-closed-loop, iterative learning control, trajectory tracking control

1. INTRODUCTION

With the rapid development of robotic technology, robots have found their industrial applications in many fields to replace a large amount of manpower. Among their applications, material handling is an important aspect, in which the Delta and SCARA robots are extensively deployed[1]. Compared to serial robots, parallel robots have received more attention thanks to their high speed, high stiffness-to-weight ratio, and low inertia,

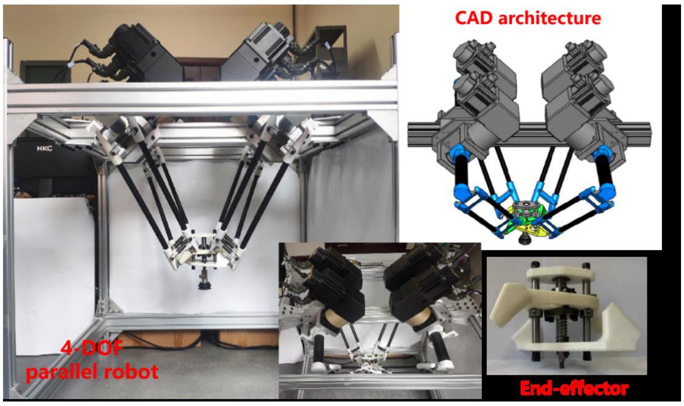


Figure 1. The prototype of the 4-dof parallel robot.

dedicated to pick-and-place operations (PPOs) with high dynamic movements. For instance, Figure 1 depicts a four-degree-of-freedom (4-dof) robot of this family suitable for PPO. Accordingly, the design of a control system for the robot under study is the focus of this work, since precise control is of importance, in particular for such a robot working with highly frequent switching of joint motions.

In the control design, classical model-free controller techniques, such as PID and PD controls, have been extensively adopted by industrial robots due to their simplicity and ease of implementation. However, these controllers are not applicable to parallel robots due to the highly nonlinear coupled characteristics[2]. In this light, some control methods, such as torque feedforward control[3], computed torque control[4], sliding mode control[5,6], etc, have been proposed to improve the control quality for parallel robots. Although those methods overcome some problems, such as trajectory tracking accuracy[6], other problems (i.e, increased computational burden and requirement of an accurate dynamic model) arise. Taking the characteristics of repetitive tasks for most parallel robots into consideration, it turns out that iterative learning control (ILC) is suitable for controlling the parallel robots, as ILC can benefit robot control from the system repeatability, wherein ILC makes use of the last output motion of the robot end-effector to obtain control inputs that can track the desired trajectory repeatedly.

ILC was first proposed in 1978[7], but it did not attract the attention of researchers until 1984 because of language restrictions[8]. Over several decades, ILC has been developed and improved with numerous variants. One example is the ILC with a P-type switching surface using a proportional structure, which can effectively cope with external disturbances[9]. Compared with the sliding mode surface, this controller is able to remove the chattering in the control process. It has been used for mobile robots to improve the robustness of path tracking against the presence of initial shifts, but it introduced a large trajectory tracking error and had a poor convergence effect[10]. The D-type ILC is proposed with an initial condition algorithm[11] to specify the initial state value in each iteration automatically. However, a lot of jittering occurs in the control torque, leading to damage to the actuator and some other robotic components. Sequentially, a modified D-type ILC was designed[12] to effectively avoid the jitter and glitch for enhanced convergence accuracy, compared to the conventional D-type one. By means of the filter, another D-type ILC method with a unit-gain derivative is proposed to compensate for the unexpected high gain of the conventional derivative at high frequency, wherein the desired phase compensation can be realized within a designated frequency band.

Despite the advantages of the above-mentioned ILC methods, neither P- nor D-type learning laws can make full use of system information. In the control law, P- and D-type gains not only play a role in learning gain but also take the task of accomplishment of the feedback in the control system[13,14]. However, it is difficult to achieve the compatibility between feedback stability and learning convergence. Alternatively, PD-type ILCs are deployed in parallel robots[15]. For instance, an open-loop PD-type ILC algorithm was proposed for a class of nonlinear time-varying systems with control delay and arbitrary initial value[16]. In this manner, the learning convergence curve is not smooth, although it solves the problem of initial shift. The robustness of the controller can be ensured by designing a robust term, aiming at the control of a 3-dof permanent magnet spherical actuator[17]. Open-loop PD-type ILCs have also been applied in the Delta robot; however, the test on the controller showed that convergence requires a number of iterations and plenty of computation time, i.e., an unacceptable computational burden in real applications[18]. To speed up the convergence of the controller, the constant gain of the PD control can be changed to a time-varying one[19], but this introduces glitches during the convergence procedure. Alternatively, an adaptive controller can be integrated, where the controller gain is defined as a function of the number of iterations[20]; sequentially, both the position and velocity tracking errors can be monotonically and rapidly reduced. In addition, to realize the automatic tuning of a controller, a method with generalization capabilities is proposed in[21] that can effectively tune the parameters to improve the trajectory tracking accuracy for robots. Besides, ILC can also be applied in repetitive rehabilitation training[22], in which a high-order ILC can improve the transient performance and decrease the steady-state error, compared to traditional PID controllers. Since ILC is equivalent to an integrator along with the iterations, it is sensitive to external disturbances[23]. The focus of this work is the design of an ILC considering disturbances for high-speed parallel robots for a pick-and-place application.

In the practical application of industrial robots, classical PD control is still the mainstream algorithm, and studies on the iterative learning theory applied to control of parallel robots have not been extensively reported. Consequently, the present work is to illustrate the effectiveness and feasibility of such algorithms for parallel robots. In this paper, an open-closed-loop PD-type ILC method is proposed and illustrated with a parallel robot producing Schönflies motion. The proposed ILC law consists of classical PD control and ILC. The iterative learning term can be regarded as feedforward compensation, which can use the information stored in the last movement. The PD control part belongs to the feedback item and performs real-time compensation. The controller convergence is proved based on Q operator theory, and the tracking performance is tested by tracking a pick-and-place trajectory and compared with the classical D-ILC controller. Moreover, different trajectories and working frequencies are selected to verify the effectiveness of the controller.

2. ROBOT STRUCTURE AND DYNAMIC MODEL

Figure 2 depicts the detailed CAD model of the robot shown in Figure 1, which is composed of a mounting frame, a screw-pair-based moving platform, and four identical limbs. Each limb consists of a big (inner) arm and a small (outer) arm. A drive motor and a reducer are installed on the rotating shaft of the big arm. The outer arm is composed of two carbon fiber rods in a π-shape. The inner and outer arms are connected by ball joints, as well as the connection between the outer arm and the mobile platform. The mobile platform can be split into two subparts, i.e., upper and lower sub-platforms. Through the helix joint, the rotation in the vertical direction of the end-effector can be generated by the differential motion of the two sub-platforms.

The kinematics and dynamics of the robot have been well documented in the previous work[24], which is revisited by skipping the details. When ignoring un-modeled errors and external disturbance, the dynamic model of the robot can be expressed as:

$$\vec{\tau} = \boldsymbol{M}(\vec{q})\ddot{\vec{q}} + \boldsymbol{C}(\vec{q},\dot{\vec{q}})\dot{\vec{q}} + \vec{\boldsymbol{G}}(\vec{q}) \tag{1}$$

with

$$\boldsymbol{M}(\vec{q}) = \boldsymbol{J}_{\mathrm{low}}^{\mathrm{T}}\boldsymbol{M}_{\mathrm{p,low}}\boldsymbol{J}_{\mathrm{low}} + \boldsymbol{J}_{\mathrm{up}}^{\mathrm{T}}\boldsymbol{M}_{\mathrm{p,up}}\boldsymbol{J}_{\mathrm{up}} + \boldsymbol{I}_{\mathrm{b}} \tag{2}$$

$$\boldsymbol{C}(\vec{q},\dot{\vec{q}}) = \boldsymbol{J}_{\mathrm{low}}^{\mathrm{T}}\boldsymbol{M}_{\mathrm{p,low}}\dot{\boldsymbol{J}}_{\mathrm{low}} + \boldsymbol{J}_{\mathrm{up}}^{\mathrm{T}}\boldsymbol{M}_{\mathrm{p,up}}\dot{\boldsymbol{J}}_{\mathrm{up}} \tag{3}$$

$$\vec{\boldsymbol{G}}(\vec{q}) = -\boldsymbol{J}_{\mathrm{low}}^{\mathrm{T}}\boldsymbol{M}_{\mathrm{p,low}}\vec{\boldsymbol{g}} - \boldsymbol{J}_{\mathrm{up}}^{\mathrm{T}}\boldsymbol{M}_{\mathrm{p,up}}\vec{\boldsymbol{g}} - \boldsymbol{M}_{\mathrm{b}}\vec{\boldsymbol{g}} \tag{4}$$

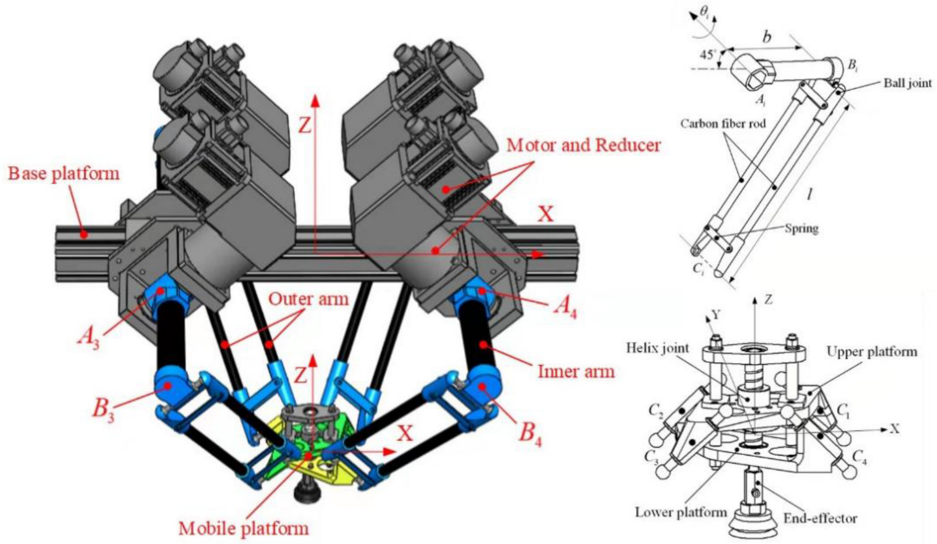


Figure 2. CAD model of the 4-dof robot with a revolute-spherical-spherical limb and a screw pair-based mobile platform.

Table 1. Geometric and dynamic parameters of the robot

Parameters	value
Length of inner arm	0.296 m
Length of outer arm	0.600 m
Mass of upper platform	0.855 kg
Mass of lower platform	1.080 kg
Mass of inner arm	0.842 kg
Mass of outer arm	0.073 kg

where $\vec{\tau} \in R^4$ is the driving torque and $\dot{\vec{q}}, \ddot{\vec{q}} \in R^4$ represent the joint angular velocity and acceleration, respectively. Moreover, $\boldsymbol{M}(\vec{q}) \in R^{4\times4}$ is the inertia matrix, $\boldsymbol{C}(\vec{q}, \dot{\vec{q}}) \in R^{4\times4}$ is a vector resulting from Coriolis and centrifugal forces, $\vec{G}(\vec{q}) \in R^4$ represents gravity, and $\boldsymbol{I}_\mathrm{b}$ is the moment of inertia of inner arms. Jacobians $\boldsymbol{J}_\mathrm{up}$ and $\boldsymbol{J}_\mathrm{down}$ relate the motion of the upper and lower sub-platforms to the actuated joints, while $\dot{\boldsymbol{J}}_\mathrm{up}$ and $\dot{\boldsymbol{J}}_\mathrm{down}$, respectively, represent their derivatives with respect to time. In addition, $\boldsymbol{M}_\mathrm{b}$, $\boldsymbol{M}_\mathrm{p,up}$, and $\boldsymbol{M}_\mathrm{p,down}$ are the mass matrices of the inner arm and the upper and lower sub-platforms. The detailed modeling procedure can be found in Ref[24]. The main geometric and dynamic parameters of the parallel robot are listed in Table 1.

3. ITERATIVE LEARNING CONTROLLER DESIGN

Prior to the ILC design for the robot, the following properties generalized to the robotic manipulators are considered.

Property 1. The inertia matrix is bounded and positive definite, thus $\exists\delta > 0, \zeta > 0$ satisfies the following inequalities:

$$0 < \delta < \|\boldsymbol{M}(\vec{q}_k, t)\| < \zeta \tag{5}$$

Property 2. The inertia matrix satisfies the global Lipschitz condition; therefore, a positive constant L exists that meets:

$$\|\boldsymbol{M}(\vec{q}_k, t) - \boldsymbol{M}(\vec{q}_{k-1}, t)\| \le L\|\vec{q}_k(t) - \vec{q}_{k-1}(t)\| \tag{6}$$

where k represents the number of iterations and $\vec{q}$ is the angular displacement of the joint.

Property 3. Coriolis, centrifugal, and gravitational force matrices meet the equation $\boldsymbol{C}(\vec{q}_k, \dot{\vec{q}}_k)\dot{\vec{q}}_d + \vec{G}(\vec{q}_k) =$

$\varphi(\vec{q}_k, \dot{\vec{q}}_k)\vec{\gamma}_k(t)$, where $\varphi(\vec{q}_k, \dot{\vec{q}}_k) \in R^{n\times m}$ is a regression matrix and $\vec{\gamma}_k(t) \in R^{m\times 1}$ is a vector of unknown parameters regarding the robot.

Moreover, the following reasonable assumptions are made.

Assumption 1. The system can meet the alignment condition, i.e., $\vec{q}_k(0) = \vec{q}_d(0), \dot{\vec{q}}_k(0) = \dot{\vec{q}}_d(0)$. The desired joint position trajectory, namely, $\vec{q}_d$, and its nth derivatives are bounded, namely, $\forall t \in [0, T], \forall k \in Z_+$.

Assumption 2. The external disturbance of the robot is bounded and is subject to a positive constant:

$$\sup \|\vec{d}_k(t)\| \leq l \tag{7}$$

In view of the nonlinear time-varying robotic system with repetitive work over a finite interval time $t \in [0, T]$, an open-closed loop PD-ILC law is designed. This algorithm belongs to the feedback–feedforward control law, which can make full use of the effective information stored in the system for learning and can ensure that the output variables converge to the bounded threshold of desired values.

The specific expression is written as follows:

$$\vec{\tau}_{k+1}(t) = \vec{\tau}_k(t) + \vec{\tau}_{\text{fore}} + \vec{\tau}_{\text{back}} \tag{8}$$

where $\vec{\tau}$ is the driving torque and k is the number of iterations. Moreover, $\vec{\tau}_{\text{fore}}$ is the feedforward control input, written as:

$$\vec{\tau}_{\text{fore}} = \boldsymbol{L}_p \vec{e}_k(t) + \boldsymbol{L}_d \dot{\vec{e}}_k(t) \tag{9}$$

where $\boldsymbol{L}_p, \boldsymbol{L}_d$ are symmetric positive definite gain matrices for the feedforward control and $\vec{e}_k = \vec{q}_k - \vec{q}_d$ and $\dot{\vec{e}}_k = \dot{\vec{q}}_k - \dot{\vec{q}}_d$ represent the joint errors in terms of angular displacement and angular velocity, respectively, in the kth iteration.

The feedback control $\vec{\tau}_{\text{back}}$ takes the following form:

$$\vec{\tau}_{\text{back}} = (1-\alpha)\boldsymbol{L}_p \vec{e}_{k+\mathbf{1}}(t) + (1-\beta)\boldsymbol{L}_d \dot{\vec{e}}_{k+\mathbf{1}}(t) \tag{10}$$

where α and β are gain coefficients of the controller.

The scheme of the proposed controller is displayed in Figure 3. It can be seen that the information obtained in the kth iteration can be regarded as the feedforward part. The current joint errors, namely, the information obtained in the $(k+1)$th iteration, constitute the feedback part of the control law. Under the condition that the control target and external environment remain unchanged, the target task is repeatedly executed, and the response of the system is identical to the feedforward information. When the system deviates from the desired trajectory, the feedback term will compensate the motion errors.

4. CONVERGENCE ANALYSIS OF THE CONTROLLER

To prove the convergence of proposed controller, the following two lemmas are introduced as the fundamentals.

Lemma 1. With $\forall \vec{x}, \vec{y} \in C_r[0, T], t \in [0, T]$, assuming that the operator $\vec{Q} : C_r[0, T] \rightarrow C_r[0, T]$ meets global Lipschitz condition, one obtains the following two outcomes.

(1) For $\forall \vec{y} \in C_r[0, T]$, there is a unique $\vec{x} \in C_r[0, T]$ that holds:

$$\vec{x}(t) + \vec{Q}(\vec{x})(t) = \vec{y}(t), \ \forall t \in [0, T] \tag{11}$$

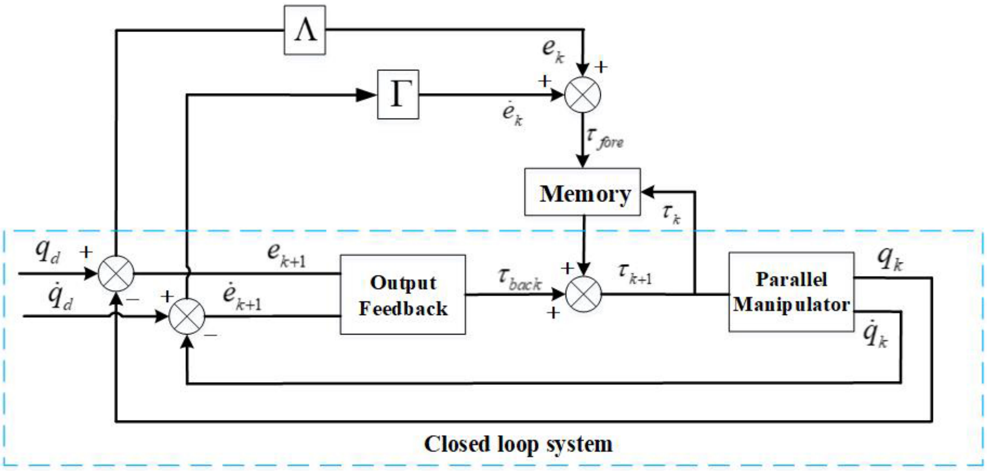


Figure 3. Scheme of open-closed-loop PD-type ILC system.

(2) Defining the operator $\vec{\bar{\boldsymbol{Q}}}$ yields

$$\vec{\bar{\boldsymbol{Q}}}(\vec{\boldsymbol{y}})(t) = \vec{\boldsymbol{Q}}(\vec{\boldsymbol{x}})(t),\ \forall \vec{\boldsymbol{y}} \in C_r[0,T] \tag{12}$$

where $\vec{\boldsymbol{x}} \in C_r[0,T]$ is the only solution to the first outcome, and there exists a constant $M_1 > 0$ subject to:

$$\|\vec{\bar{\boldsymbol{Q}}}(\vec{\boldsymbol{x}})(t)\| \le M_1 \left(q + \int_0^t \|\vec{\boldsymbol{y}}(s)\| \mathrm{d}s \right) \tag{13}$$

Lemma 2. Assuming that the sequence $\{b_k\}_{k\ge 0}$, $b_k \ge 0$, converges to zero, the operator $\vec{\boldsymbol{Q}}_k : C_r[0,T] \to C_r[0,T]$ will meet

$$\|\vec{\boldsymbol{Q}}_k(\vec{\boldsymbol{x}})(t)\| \le M \left(b_k + \int_0^t \|\vec{\boldsymbol{x}}(s)\| \mathrm{d}s + \sigma \right) \tag{14}$$

where $\sigma > 0$ and $M \ge 1$ are constants. Assuming that $\boldsymbol{P}(t)$ is a $r \times r$ continuous function matrix, the operator $\vec{\boldsymbol{P}} : C_r[0,T] \to C_r[0,T]$ satisfies $\vec{\boldsymbol{P}}(\vec{\boldsymbol{x}})(t) = \boldsymbol{P}(t)\vec{\boldsymbol{x}}(t)$. If $\rho < 1$, ρ being the spectral radius of $\vec{\boldsymbol{P}}$, for $\forall t \in [0,T]$, there exists

$$\lim_{n\to\infty} (\vec{\boldsymbol{P}} + \vec{\boldsymbol{Q}}_{\mathrm{n}})(\vec{\boldsymbol{P}} + \vec{\boldsymbol{Q}}_{n-1}) \cdots (\vec{\boldsymbol{P}} + \vec{\boldsymbol{Q}}_0)(\vec{\boldsymbol{x}})(t) = 0$$

For the parallel robot under study, the state variables $\vec{\boldsymbol{X}} = [\vec{\boldsymbol{x}}_1, \vec{\boldsymbol{x}}_2]^{\mathrm{T}}_{8\times 1}$ are defined below:

$$\begin{cases} \vec{\boldsymbol{x}}_1 = \vec{\boldsymbol{q}} \\ \vec{\boldsymbol{x}}_2 = \dot{\vec{\boldsymbol{q}}} \end{cases} \tag{15}$$

Accordingly, the variable $\vec{\boldsymbol{\phi}}(t, \vec{\boldsymbol{X}})_{4\times 1} = -\boldsymbol{M}^{-1}(\vec{\boldsymbol{q}})(\boldsymbol{C}(\vec{\boldsymbol{q}}, \dot{\vec{\boldsymbol{q}}})\dot{\vec{\boldsymbol{q}}} + \vec{\boldsymbol{G}}(\vec{\boldsymbol{q}}))$ can be defined; thus, the dynamic model of the system can be expressed as:

$$\dot{\vec{\boldsymbol{X}}} = \begin{bmatrix} \dot{\vec{\boldsymbol{x}}}_1 \\ \dot{\vec{\boldsymbol{x}}}_2 \end{bmatrix} = \begin{bmatrix} \dot{\vec{\boldsymbol{q}}} \\ \vec{\boldsymbol{\phi}}(t, \vec{\boldsymbol{X}}) \end{bmatrix} + \begin{bmatrix} 0 \\ \boldsymbol{M}^{-1}(\vec{\boldsymbol{q}})\vec{\tau}(t) \end{bmatrix} \tag{16}$$

As a consequence, the state equation of the robot can be obtained:

$$\begin{cases} \dot{\vec{\boldsymbol{X}}} = \vec{\boldsymbol{\Phi}}(t, \vec{\boldsymbol{X}}) + \boldsymbol{B}(\vec{\boldsymbol{q}}, \dot{\vec{\boldsymbol{q}}})\vec{\tau}(t) \\ \vec{\boldsymbol{Y}} = \boldsymbol{C}\vec{\boldsymbol{X}} \end{cases} \tag{17}$$

where $\vec{\boldsymbol{\Phi}}(t,\vec{\boldsymbol{X}})_{8\times1} = [\dot{\vec{\boldsymbol{q}}}\vec{\boldsymbol{\phi}}(t,\vec{\boldsymbol{X}})]^{\mathrm{T}}_{8\times1}$, $\boldsymbol{B}(\vec{\boldsymbol{q}},\dot{\vec{\boldsymbol{q}}})_{8\times4} = [0,\boldsymbol{M}^{-1}(\vec{\boldsymbol{q}})]^{\mathrm{T}}_{8\times4}$, and $\boldsymbol{C} = [0,\boldsymbol{I}]^{\mathrm{T}}_{8\times4}$.

For the nonlinear system of Equation (17), based on the ILC law in Equation (8), if the system can meet the following condition,

$$\rho([\boldsymbol{I}+\boldsymbol{L}_d\boldsymbol{C}(t)\boldsymbol{B}(t)]^{-1}[\boldsymbol{I}-\boldsymbol{L}_d\boldsymbol{C}(t)\boldsymbol{B}(t)]) < 1, t\in[0,T] \tag{18}$$

the trajectory tracking error of the dynamic system converges to a certain small range with the increasing iterations.

Let the system state, output, and input errors be set as:

$$\begin{cases}\delta\vec{\boldsymbol{X}}_k(t) = \vec{\boldsymbol{X}}_d(t)-\vec{\boldsymbol{X}}_k(t)\\ \delta\vec{\boldsymbol{Y}}_k(t) = \vec{\boldsymbol{Y}}_d(t)-\vec{\boldsymbol{Y}}_k(t)\\ \delta\vec{\boldsymbol{\tau}}_k(t) = \vec{\boldsymbol{\tau}}_d(t)-\vec{\boldsymbol{\tau}}_k(t)\end{cases} \tag{19}$$

Defining the variable $\vec{\boldsymbol{\Phi}}_1(\vec{\boldsymbol{X}}(t),t) = \vec{\boldsymbol{\Phi}}(\vec{\boldsymbol{X}}_d(t),t)-\vec{\boldsymbol{\Phi}}(\vec{\boldsymbol{X}}_d(t)-\vec{\boldsymbol{X}}(t),t)$, the following inequalities can be obtained by Lipschitz condition:

$$\begin{cases}\|\vec{\boldsymbol{\Phi}}_1(\vec{\boldsymbol{X}}(t),t)\| \le L_1\\ \|\vec{\boldsymbol{\Phi}}_1(\vec{\boldsymbol{X}}_1(t),t)-\vec{\boldsymbol{\Phi}}_1(\vec{\boldsymbol{X}}_2(t),t)\| \le L_2\|\vec{\boldsymbol{X}}_1(t)-\vec{\boldsymbol{X}}_2(t)\|\end{cases} \tag{20}$$

Combining Equations (17) and (19) results in

$$\begin{cases}\delta\dot{\vec{\boldsymbol{X}}}_k(t) = \vec{\boldsymbol{\Phi}}_1(\delta\vec{\boldsymbol{X}}_k(t),t)+B(t)\delta\vec{\boldsymbol{\tau}}_k(t)\\ \delta\vec{\boldsymbol{Y}}_k(t) = \boldsymbol{C}(t)\delta\vec{\boldsymbol{X}}_k(t)\\ \delta\dot{\vec{\boldsymbol{Y}}}_k(t) = \dot{\boldsymbol{C}}(t)\delta\vec{\boldsymbol{X}}_k(t)+\boldsymbol{C}(t)\delta\dot{\vec{\boldsymbol{X}}}_k(t)\end{cases} \tag{21}$$

with

$$\begin{aligned}\delta\vec{\boldsymbol{\tau}}_{k+1}(t) = {}&\delta\vec{\boldsymbol{\tau}}_k(t)-\boldsymbol{L}_p\delta\vec{\boldsymbol{Y}}_k(t)-\boldsymbol{L}_p\delta\vec{\boldsymbol{Y}}_{k+1}(t)-\boldsymbol{L}_d(t)\delta\dot{\vec{\boldsymbol{Y}}}_k(t)\\ &-\boldsymbol{L}_d(t)\delta\dot{\vec{\boldsymbol{Y}}}_{k+1}(t)+\alpha\boldsymbol{L}_p\delta\vec{\boldsymbol{Y}}_{k+1}(t)+\beta\boldsymbol{L}_d(t)\delta\dot{\vec{\boldsymbol{Y}}}_{k+1}(t)\end{aligned} \tag{22}$$

Substituting Equation (21) into Equation (22) yields

$$\begin{aligned}\delta\vec{\boldsymbol{\tau}}_{k+1}(t) = {}&\delta\vec{\boldsymbol{\tau}}_k(t)-\boldsymbol{L}_p\boldsymbol{C}(t)\delta\vec{\boldsymbol{X}}_k(t)-(1-\alpha)\boldsymbol{L}_p\boldsymbol{C}(t)\delta\vec{\boldsymbol{X}}_{k+1}(t)\\ &-\boldsymbol{L}_d(\dot{\boldsymbol{C}}(t)\delta\vec{\boldsymbol{X}}_k(t)+\boldsymbol{C}(t)\vec{\boldsymbol{\Phi}}_1(t,\delta\vec{\boldsymbol{X}}_k(t))+\boldsymbol{C}(t)\boldsymbol{B}(t)\delta\vec{\boldsymbol{\tau}}_k(t))\\ &-(1-\beta)\boldsymbol{L}_d(\dot{\boldsymbol{C}}(t)\delta\vec{\boldsymbol{X}}_{k+1}(t)+\boldsymbol{C}(t)\vec{\boldsymbol{\Phi}}_1(t,\delta\vec{\boldsymbol{X}}_{k+1}(t))+\boldsymbol{C}(t)\boldsymbol{B}(t)\delta\vec{\boldsymbol{\tau}}_{k+1}(t))\end{aligned} \tag{23}$$

Let us define the operator $\vec{\boldsymbol{Q}}_k,\vec{\boldsymbol{G}}_k,\vec{\boldsymbol{P}}_k : C_r[0,T]\longrightarrow C_r[0,T]$ as follows:

$$\begin{cases}\vec{\boldsymbol{Q}}_k(\vec{\boldsymbol{\tau}})(t) = \boldsymbol{L}_p\boldsymbol{C}(t)\vec{\boldsymbol{X}}(t)+\boldsymbol{L}_d\dot{\boldsymbol{C}}(t)\vec{\boldsymbol{X}}(t)+\boldsymbol{L}_d\boldsymbol{C}(t)\vec{\boldsymbol{\Phi}}_1(\vec{\boldsymbol{X}}(t),t)\\ \vec{\boldsymbol{G}}_{k+1}(\vec{\boldsymbol{\tau}})(t) = [\boldsymbol{I}+(1-\boldsymbol{\beta})\boldsymbol{L}_d\boldsymbol{C}(t)\boldsymbol{B}(t)]^{-1}(1-\boldsymbol{\beta})\vec{\boldsymbol{Q}}_{k+1}(\vec{\boldsymbol{\tau}})(t)\\ \vec{\boldsymbol{P}}_k(\vec{\boldsymbol{\tau}})(t) = -[\boldsymbol{I}+(1-\boldsymbol{\beta})\boldsymbol{L}_d\boldsymbol{C}(t)\boldsymbol{B}(t)]^{-1}\alpha\vec{\boldsymbol{Q}}_k(\vec{\boldsymbol{\tau}})(t)\end{cases} \tag{24}$$

According to the authors of Ref[23], $\vec{\boldsymbol{Q}}_k, \vec{\boldsymbol{G}}_k, \vec{\boldsymbol{P}}_k$ should meet the conditions of Lemma 1:

$$\begin{cases} \|\vec{\boldsymbol{Q}}_k(\vec{\boldsymbol{\tau}})(t)\| \leq M_Q(\|\boldsymbol{x}(0)\| + \int_0^t \|\vec{\boldsymbol{\tau}}(s)\|\mathrm{d}s) \\ \|\vec{\boldsymbol{G}}_{k+1}(\vec{\boldsymbol{\tau}})(t)\| \leq M_{\mathrm{G}}(\|\boldsymbol{x}(0)\| + \int_0^t \|\vec{\boldsymbol{\tau}}(s)\|\mathrm{d}s) \\ \|\vec{\boldsymbol{P}}_k(\vec{\boldsymbol{\tau}})(t)\| \leq M_P(\|\boldsymbol{x}(0)\| + \int_0^t \|\vec{\boldsymbol{\tau}}(s)\|\mathrm{d}s) \end{cases} \tag{25}$$

Let us define the operator $\vec{\boldsymbol{S}}, \vec{\boldsymbol{W}}_k : C_r[0,T] \longrightarrow C_r[0,T]$ below:

$$\begin{cases} \vec{\boldsymbol{S}}(\vec{\boldsymbol{\tau}})(t) = [\boldsymbol{I} + (1-\beta)\boldsymbol{L}_d\boldsymbol{C}(t)\boldsymbol{B}(t)]^{-1}[I - \alpha\boldsymbol{L}_d\boldsymbol{C}(t)\boldsymbol{B}(t)]\vec{\boldsymbol{\tau}}(t) \\ \vec{\boldsymbol{W}}_k(\vec{\boldsymbol{\tau}})(t) = (\vec{\boldsymbol{P}}_k + \vec{\boldsymbol{S}})(\vec{\boldsymbol{\tau}})(t) \end{cases} \tag{26}$$

Equation (23) can be rewritten as:

$$\delta\vec{\boldsymbol{\tau}}_{k+1}(t) + \vec{\boldsymbol{G}}_{k+1}(\delta\vec{\boldsymbol{\tau}}_{k+1}(t))(t) = \vec{\boldsymbol{W}}_k(\delta\vec{\boldsymbol{\tau}}_k(t))(t) \tag{27}$$

Since $\vec{\boldsymbol{G}}_{k+1}(\vec{\boldsymbol{\tau}})(t)$ can meet Lemma 1, the following operators can be defined:

$$\begin{cases} \vec{\bar{\boldsymbol{G}}}_{k+1}(\vec{\boldsymbol{Y}})(t) = \vec{\boldsymbol{G}}_{k+1}(\vec{\boldsymbol{\tau}})(t) \\ \vec{\boldsymbol{Z}}_{k+1}(\vec{\boldsymbol{Y}})(t) = \vec{\bar{\boldsymbol{G}}}_{k+1}(\vec{\boldsymbol{Y}})(t) \end{cases} \tag{28}$$

where $\vec{\boldsymbol{\tau}}(t) + \vec{\boldsymbol{G}}_{k+1}(\vec{\boldsymbol{\tau}})(t) = \vec{\boldsymbol{Y}}(t), \forall\vec{\boldsymbol{Y}}(t) \in C_r[0,T]$. Comparing with Equation (27), the following relationship can be obtained:

$$\begin{cases} \vec{\boldsymbol{Z}}_{k+1}(\delta\vec{\boldsymbol{\tau}}_k)(t) = -\vec{\bar{\boldsymbol{G}}}_{k+1}(\vec{\boldsymbol{W}}_k(\delta\vec{\boldsymbol{\tau}}_k))(t) \\ \vec{\bar{\boldsymbol{G}}}_{k+1}(\vec{\boldsymbol{W}}_k(\delta\vec{\boldsymbol{\tau}}_k))(t) = \vec{\boldsymbol{G}}_{k+1}(\delta\vec{\boldsymbol{\tau}}_{k+1})(t) \end{cases} \tag{29}$$

From Lemma 1, one obtains

$$\|\vec{\bar{\boldsymbol{G}}}_{k+1}(\vec{\boldsymbol{W}}_k(\delta\vec{\boldsymbol{\tau}}_k))(t)\| \leq M_{\bar{G}}(\|\delta\vec{\boldsymbol{X}}_{k+1}(0)\| + \int_0^t \|\vec{\boldsymbol{W}}_k(\delta\vec{\boldsymbol{\tau}}_k)(s)\|\mathrm{d}s \tag{30}$$

From Equations (24), (26), and (30), the following equation can be derived

$$\|\vec{\boldsymbol{Z}}_{k+1}(\delta\vec{\boldsymbol{\tau}}_k)(t)\| \leq M_Z(\|\delta\vec{\boldsymbol{X}}_{k+1}(0)\| + \|\delta\vec{\boldsymbol{X}}_k(0)\| + \int_0^t \|\delta\vec{\boldsymbol{\tau}}_k(s)\|\mathrm{d}s \tag{31}$$

where $M_Z = M_{\bar{G}} \cdot \max(M_V, 1)$.

Let us define the operator $\vec{\boldsymbol{J}}_k : C_r[0,T] \longrightarrow C_r[0,T]$ as follows:

$$\vec{\boldsymbol{J}}_k(\vec{\boldsymbol{\tau}})(t) = (\vec{\boldsymbol{P}}_k + \vec{\boldsymbol{Z}}_{k+1})(\vec{\boldsymbol{\tau}})(t) \tag{32}$$

Equation (27) can be expressed accordingly as:

$$\delta\vec{\boldsymbol{\tau}}_{k+1}(t) = \vec{\boldsymbol{Z}}_{k+1}(\delta\vec{\boldsymbol{\tau}}_k(t))(t) + \vec{\boldsymbol{W}}_k(\delta\vec{\boldsymbol{\tau}}_k(t))(t) = (\vec{\boldsymbol{Z}}_{k+1} + \vec{\boldsymbol{P}}_k + \vec{\boldsymbol{S}})(\delta\vec{\boldsymbol{\tau}}_k)(t) \tag{33}$$

Taking the norm on both sides of Equation (32) and substituting the inequalities in Equations (25) and (31) into Equation (32) leads to

$$\|\vec{\boldsymbol{J}}_k(\delta\vec{\boldsymbol{\tau}}_k)(t)\| \leq \|\vec{\boldsymbol{P}}_k(\delta\vec{\boldsymbol{\tau}}_k)(t)\| + \|\vec{\boldsymbol{Z}}_{k+1}(\delta\vec{\boldsymbol{\tau}}_k)(t)\|$$

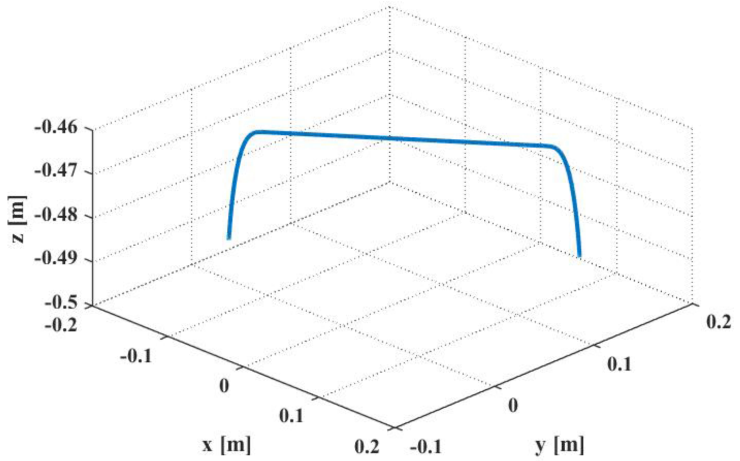


Figure 4. Test trajectory of the pick-and-place operation.

$$
\begin{aligned}
&\leq M_P(\|\delta\vec{X}(0)\| + \int_0^t \|\delta\vec{\tau}(s)\|\mathrm{d}s) + M_Z(\|\delta\vec{X}_{k+1}(0)\| + \|\delta\vec{X}_k(0)\| + \int_0^t \|\delta\vec{\tau}_k(s)\|\mathrm{d}s \\
&\leq \max(1, M_P + M_Z)\left(\|\delta\vec{X}_{k+1}(0)\| + \|\delta\vec{X}_k(0)\| + \int_0^t \|\delta\vec{\tau}_k(s)\|\mathrm{d}s\right)
\end{aligned}
\tag{34}
$$

Finally, Equation (33) can be expressed as:

$$
\delta\vec{\tau}_{k+1}(t) = (\vec{S} + \vec{J}_k)(\delta\vec{\tau}_k)(t) = (\vec{S} + \vec{J}_k)(\vec{S} + \vec{J}_{k-1})\cdots(\vec{S} + \vec{J}_0)(\delta\vec{\tau}_0)(t)
\tag{35}
$$

In accordance with Lemma 2, if $\rho < 1$, ρ being the spectral radius of $\vec{S}$, for a finite interval time $t \in [0, T]$, $\lim_{k\to\infty}\delta\vec{\tau}_{k+1}(t) = 0$ exists.

5. EVALUATION OF CONTROLLER DESIGN

5.1. Controller performance analysis

For the parallel robots designed for PPOs, the controller is evaluated along with an industrial gate-shaped trajectory of 25 × 305 × 25 mm[6], as shown in Figure 4, and the working frequency is set to 2 Hz, i.e., 0.25 s per single journey. To evaluate the performance of the proposed control law, the classical D-ILC is used as a comparison method, and the following three indices, i.e., maximum absolute error ($MaxE$), absolute mean error (MAE), and root-mean-squared error ($RMSE$), are defined:

$$
\begin{cases}
MaxE = \max(|q_i - q_{id}|) \\
MAE = \frac{1}{m}\sum_{i=1}^{m}|q_i - q_{id}| \\
RMSE = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(q_i - q_{id})^2}
\end{cases}
\tag{36}
$$

where m stands for the number of samples collected from one iteration, q_i is the actual angular displacement of the ith joint, and q_{id} is the expected angular displacement.

For the nonlinear time-varying system of the robot described by Equation (17), the controller parameters $\alpha = 1.1$, $\beta = 1.22$, $\boldsymbol{L}_p = \mathrm{diag}([1000 \;\; 1000 \;\; 1000 \;\; 1000]$ and $\boldsymbol{L}_d = \mathrm{diag}([230 \;\; 230 \;\; 230 \;\; 230])$ are selected after multiple tunings. Upon the implementation of the two ILC laws, the comparison of the actual and expected joint displacements are shown in Figure 5, together with the trajectory tracking results displayed in Figure 6.

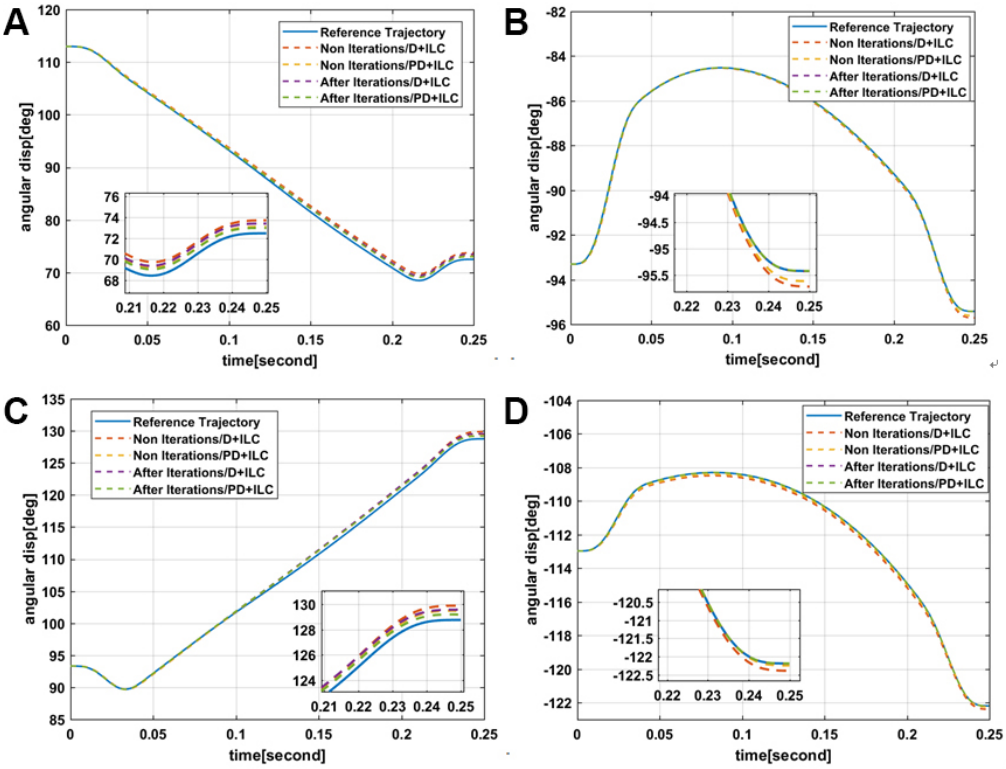


Figure 5. Comparison of the actual and expected joint displacements: (A-D) Joints 1–4.

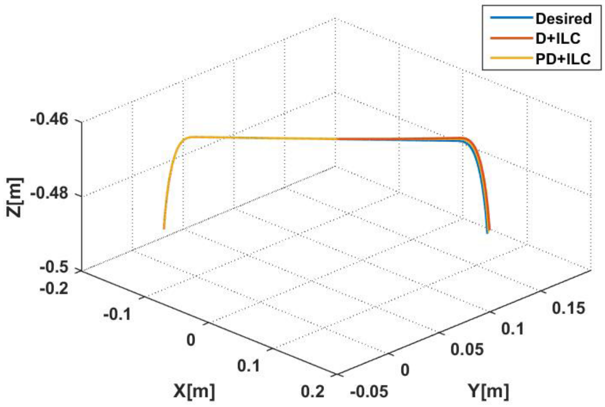


Figure 6. The trajectory tracking under D-ILC and PD-ILC.

It is observed that both ILC laws can realize trajectory tracking control, and the proposed law is superior to the D-ILC law.

Figure 7 shows the varying tracking errors of each joint. The maximum and mean tracking errors of the two controllers are given in Table 2. As shown in Figure 7, the two controllers have similar error trends. The errors of Joints 1 and 3 increase rapidly from the beginning of the rotational motion and reach the maximum values after the complete rotation, of which the maximum values are 0.94° and 0.81° for D-ILC and 0.71° and 0.61° for PD-ILC, respectively. The other two joints can achieve good performances after iterative learning, with the maximum errors approximating to zero, as shown in Table 2.

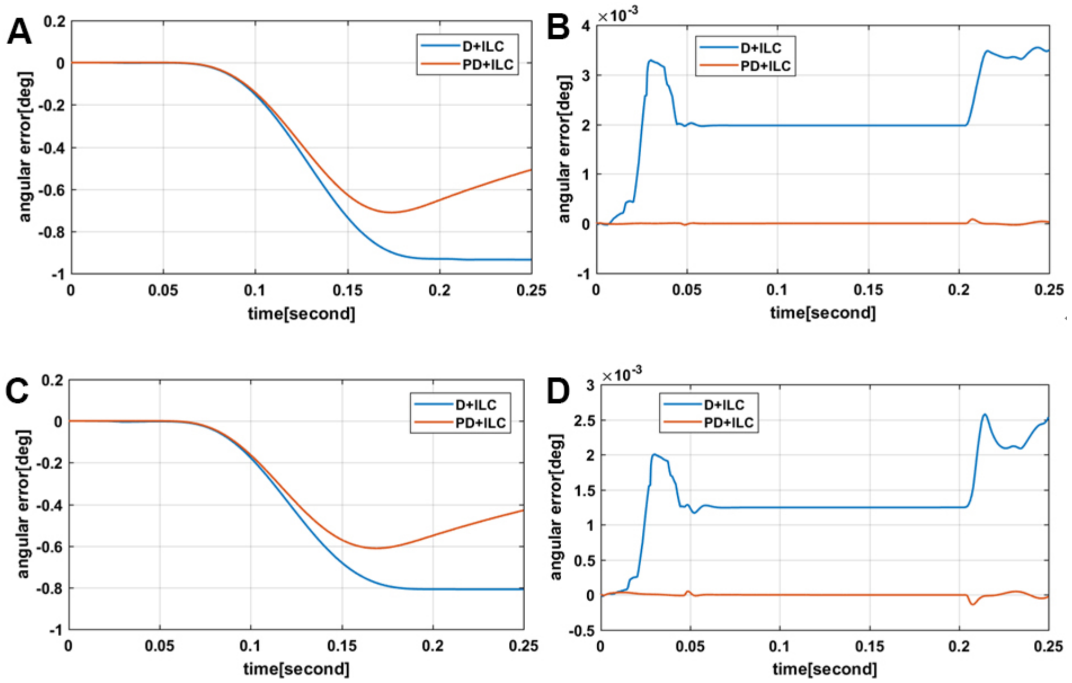


Figure 7. Trajectory tracking errors of actuated joints with D-ILC and PD-ILC laws after learning iterations: (A-D) Joints 1–4.

Table 2. The tracking errors of joints under D-ILC and PD-ILC law

	Max Error (deg)				Mean Error (deg)			
Joint i Controller	1	2	3	4	1	2	3	4
D-ILC	0.94	0.0035	0.81	0.0026	0.33	0.0018	0.30	0.0016
PD-ILC	0.71	0.0021	0.61	0.0016	0.27	0.0004	0.24	0.0003

Although the proposed control law presents superior performance compared to D-ILC, especially for Joints 2 and 4, the convergence errors of the others are still quite large. The reason lies in two aspects. On the one hand, the rotation of the robot end-effector is generated through the relative movement of the upper platform by Limbs 1 and 3, while the remaining limbs keep static. Simultaneously, the rotational motion is not continuous with the previous; therefore, the learned information cannot compensate for the errors well. On the other hand, the ILC algorithm is equivalent to an integrator along the iterative axis. It cannot guarantee that the learned information is all useful, which will lead to large errors.

Figure 8 shows the error convergence curves, where the system errors gradually converge with the increasing iterations. It can be seen that the angular displacement errors have significantly reduced after the first learning. The joint errors will become constant after the fourth iteration under the PD-ILC controller. On the contrary, there is an increase under the D-ILC law in the process of convergence.

The RMSEs for Joints 2 and 4 tend to zero from 0.0556° and 0.0952° under the PD-ILC law, while the errors under the D-ILC control law converge from 0.0874° and 0.1850° to 0.0021° and 0.0013°, respectively. The RMSEs for Joints 1 and 3 eventually converge to 0.3963° and 0.3473° for PD-ILC and 0.5180° and 0.4597° for D-ILC, respectively. It can be clearly seen that PD-ILC presents superior performance compared to the D-ILC controller.

5.2. Robustness analysis

In the real robotic application, the changes of the external environment and the existence of uncertain parameters make it difficult for the system to achieve the ideal state. For instance, the uncertain parameters of the

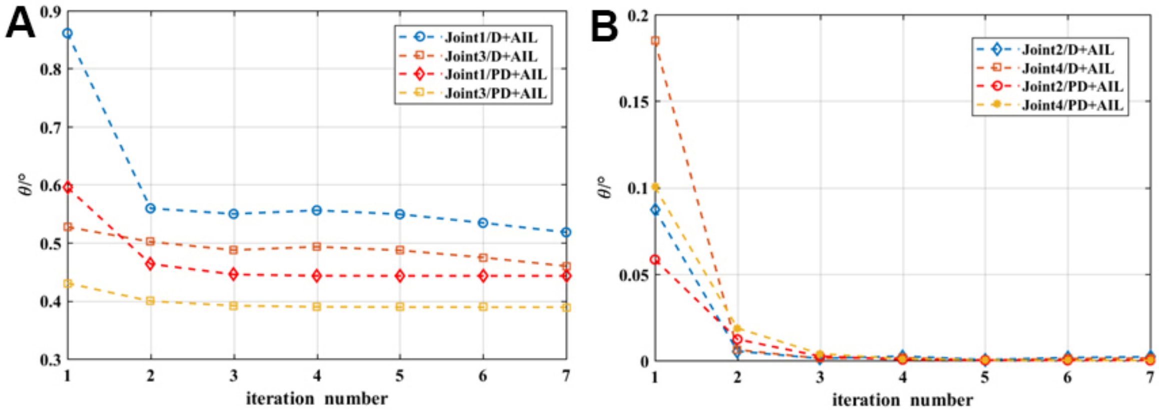


Figure 8. The varying RMSEs along with the iterations: (A) Joints 1 and 3; and (B) joints 2 and 4.

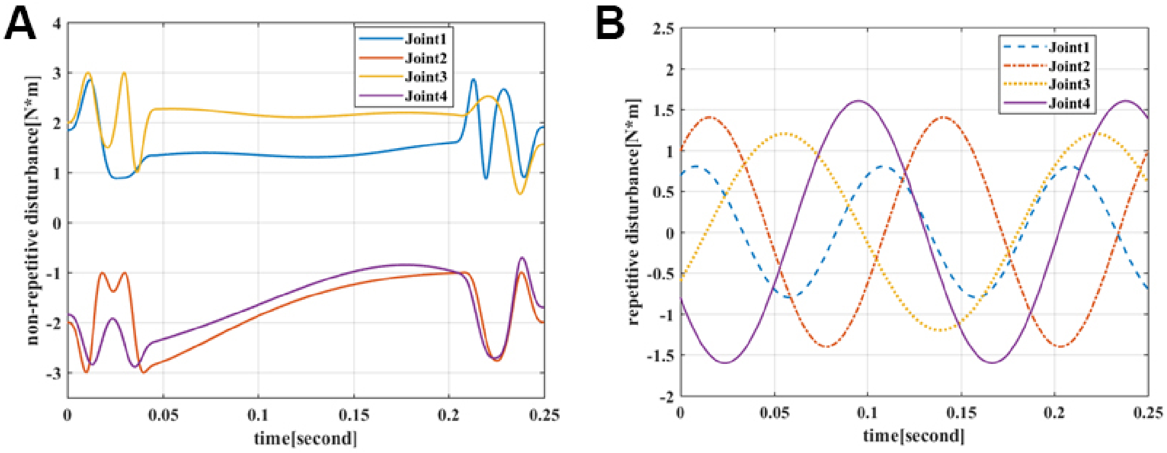


Figure 9. Repetitive (A) and non-repetitive (B) disturbance torques.

robot and the joint friction in the movement will cause interference. In view of the external environment of such a robotic system, unpredictable and random disturbances may occur; therefore, the following two forms of disturbance are defined:

$$\begin{cases} \vec{\tau}_{\mathrm{dis}} = 2\sin(\vec{q}_d) - \sin(\dot{\vec{q}}_d) \\ \vec{\tau}_{\mathrm{dis\_re}} = \lambda \sin(\alpha t + \varphi) \end{cases} \tag{37}$$

where $\vec{\tau}_{\mathrm{dis\_re}}$ represents the repetitive disturbance torque and $\vec{\tau}_{\mathrm{dis}}$ is non-repetitive disturbance torque, λ being the repetitive disturbance gain. Moreover, α and φ stand for the angular frequency and phase, respectively. Figure 9 shows the corresponding repetitive and non-repetitive disturbance torques of each joint.

Figure 10 depicts the error convergences with the increasing iterations when considering the disturbance. Compared to Figure 8, the finally converged errors of the proposed ILC are larger, compared to the error convergences without disturbance, which shows that the influence of the disturbance onto the motion accuracies of the joints cannot be ignored. The maximum and mean tracking errors with disturbance and without disturbance are given in Table 3. It is noteworthy that, when the system has external disturbances, the joint errors of the robot can still converge to a certain range after iterative learning, which indicates the robustness of the proposed control law.

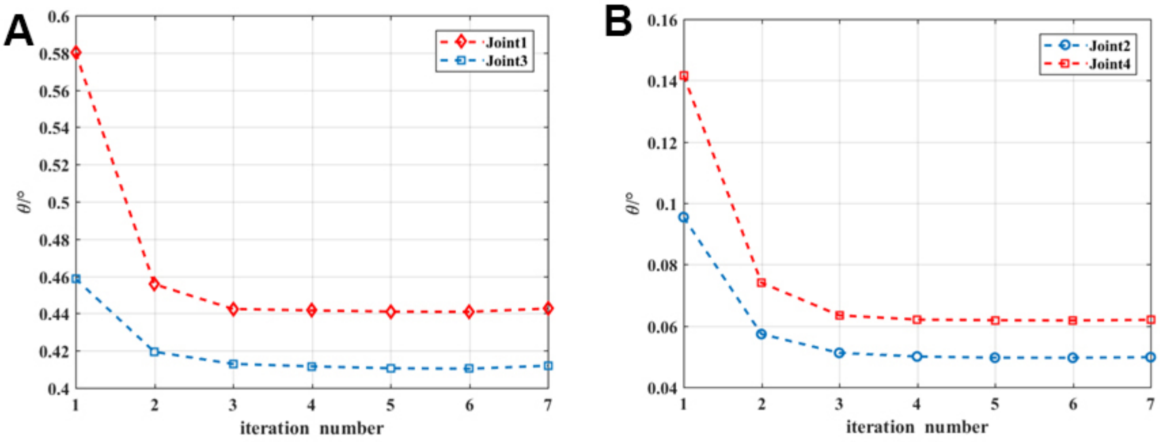


Figure 10. The varying RMSE with the increasing iterations: (A) Joints 1 and 3; and (B) joints 2 and 4.

Table 3. The tracking errors under non-disturbance and disturbance

	Max Error (deg)				Mean Error (deg)			
Joint i	1	2	3	4	1	2	3	4
Non-disturbance	0.71	0.0021	0.61	0.0016	0.27	0.0004	0.24	0.0003
Disturbance	0.76	0.087	0.68	0.089	0.32	0.039	0.30	0.057

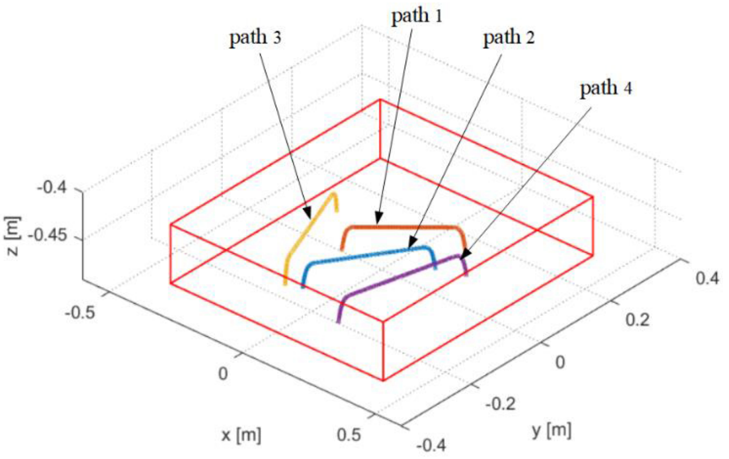


Figure 11. Different pick-and-place trajectories within the workspace.

Table 4. The tracking errors along with different paths within the workspace

	Max Error (deg)				Mean Error (deg)			
Joint i Path i	1	2	3	4	1	2	3	4
Path 1	0.71	0.0021	0.61	0.0016	0.27	0.0004	0.24	0.0003
Path 2	0.62	0.0001	0.67	0.0001	0.22	0.00003	0.24	0.00004
Path 3	0.27	0.0023	0.76	0.0015	0.069	0.0006	0.19	0.0003
Path 4	0.53	0.0009	0.33	0.0017	0.16	0.0002	0.10	0.0004

5.3. Overall performance analysis

To evaluate the overall performance of ILC in the workspace, multiple pick-and-place trajectories are selected, as displayed in Figure 11. Table 4 shows the maximum and mean tracking errors of the joints along with different paths, from which it can be seen that all the joint errors along with the selected trajectories can converge to a value after iterative learning, and the converged magnitudes are quite close.

Table 5. Results of different working frequencies with the proposed controller

	Max. Error (deg)				Mean Error (deg)			
Joint i Time i	1	2	3	4	1	2	3	4
0.25	0.71	0.0021	0.61	0.0016	0.27	0.0004	0.24	0.0003
0.15	0.78	0.0039	0.68	0.0021	0.37	0.0013	0.33	0.0007
0.50	0.58	0.0004	0.50	0.00025	0.23	0.0001	0.20	0.00008

Table 6. Results by tracking different PPO trajectories

Error Type	Joint i	4-5-6-7 th polynomial	5 th polynomial
Max. Error (deg)	Joint 1	0.7113	0.6854
	Joint 2	0.0021	0.0032
	Joint 3	0.6116	0.5875
	Joint 4	0.0016	0.0020
Mean Error (deg)	Joint 1	0.2697	0.2707
	Joint 2	0.0004	0.0013
	Joint 3	0.2404	0.2447
	Joint 4	0.0003	0.0009

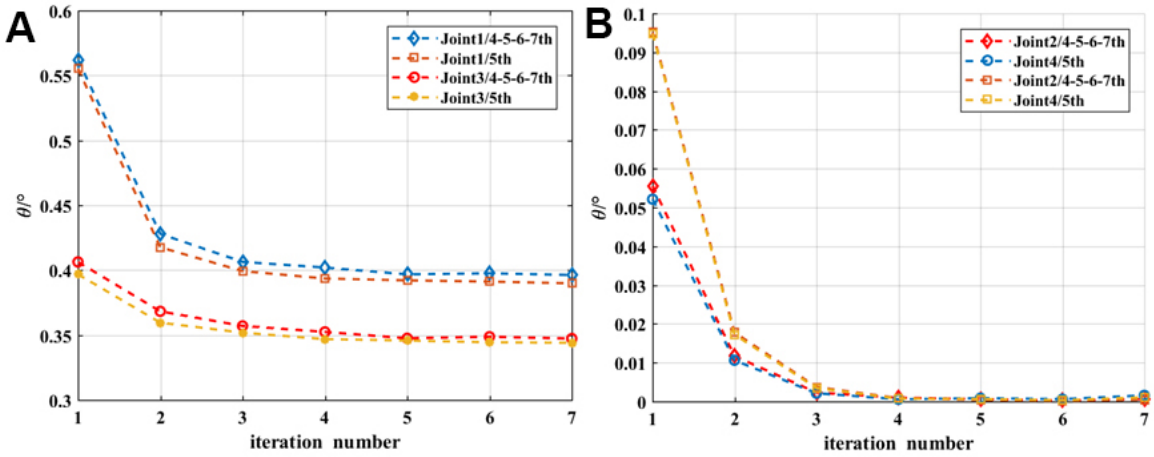


Figure 12. The varying RMSEs for different trajectories: (A) Joints 1 and 3; and (B) joints 2 and 4.

Moreover, different working frequencies and trajectories are selected to evaluate the generalization ability of the controller. The results are listed in Tables 5 and 6, respectively. Figure 12 shows the varying RMSE for different trajectories.

From the results, it can be seen that the proposed controller shows good performance under different operating frequencies and different trajectories, meaning that the proposed control law can work effectively to track different task trajectories and have good generalization capabilities.

6. CONCLUSIONS

In this work, an open-closed loop PD type iterative learning control method is proposed for parallel robots to track repetitive work trajectories, thanks to its advantages of simple implementation and practicability in industrial engineering. According to the complexity and uncertainties of the working environment, two external disturbances, i.e., repetitive and non-repetitive ones, are taken into account for the model-based control design. The designed controller is compared with the D-ILC law and evaluated along with a 4-dof parallel robot, and the results show the better performance of the PD-ILC law compared with the classical D-ILC law. The test results with and without disturbances also show the robustness in terms of the trajectory tracking errors. In addition, different working frequencies and trajectories are adopted to evaluate the generalization capabilities of the controller, and the results show that the proposed PD-ILC controller has good overall performance. The developed controller can effectively work with acceptable motion errors and computation burden from the perspective of industrial engineering, which is applicable to other high-speed parallel robots of this family. In the future, the control variables will be optimized for performance improvement.

DECLARATIONS

Authors' contributions

Conceptualization, Methodology, Software, Writing, & editing: Li Q
Software, Data curation: Liu E
Conceptualization, Review: Cui C
Conceptualization, Methodology, Review & editing, Proofreading: Wu G
All the authors approved the submitted manuscript.

Availability of data and materials

Not applicable.

Financial support and sponsorship

This work was supported by Natural Science Foundation of Liaoning Province (Grant No. 20180520028).

Conflicts of interest

The author declared that there are no conflicts of interest.

Ethical approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

Copyright

REFERENCES

1. Suri S, Jain A, Verma N, Prasertpoj N. SCARA Industrial Automation Robot, 2018 International Conference on Power Energy, Environment and Intelligent Control (PEEIC), 2018; p. 173-77.
2. Song X, Zhao Y, Jin L, Zhang P, Chen C. Dynamic feedforward control in decoupling space for a four-degree-of-freedom parallel robot. *International Journal of Advanced Robotic Systems* 2019;16:172988141882045.
3. Dang J, Ni F, Liu Y, et al. Control strategy for flexible manipulator based on feedforward compensation and fuzzy-sliding mode control. Journal of Xi'an Jiaotong University. 2011, 45(03): 75-80. Available from: https://scholar.google.com.hk/scholar?hl=zh-CN&as_sdt=0%2C5&q=Control+Strategy+for+Flexible+Manipulator+Based+on+Feedforward+Compensation+and+Fuzzy-Sliding+Mode+Control+%5BJ%5D.+Journal+of+Xi%27an+Jiaotong+University.+2011%2C+45%2803%29%3A+75-80.&btnG [Last accessed on 18 Mar 2022]
4. Dao QT, Yamamoto SI. Modified computed torque control of a robotic Orthosis for gait rehabilitation. *Annu Int Conf IEEE Eng Med Biol Soc* 2018; 2018:1719-22.
5. Su Y, Zheng C. A new nonsingular integral terminal sliding mode control for robot manipulators. *International Journal of Systems Science* 2020; 51:1418-28.
6. Wu G, Zhang X, Zhu L, Lin Z, Liu J. Fuzzy sliding mode variable structure control of a high-speed parallel PnP robot. *Mechanism and Machine Theory* 2021;162:104349.
7. Uchiyama M. Formation of high-speed motion pattern of a mechanical arm by trial. *TSICE* 1978;14:706-12.
8. Arimoto S, Kawamura S, Miyazaki F. Bettering operation of Robots by learning. *J Robotic Syst* 1984;1:123-40.

9. Long Y, Du Z, Wang W. An adaptive sliding mode-like P-type iterative learning control for robot manipulators. 2014 14th International Conference on Control, Automation and Systems (ICCAS 2014).
10. Zhao Y, Zhou F, Wang D, et al. Path-tracking of mobile robot using feedback-aided P-type iterative learning control against initial state error. *IEEE* 2017.
11. Bouakrif F. D-type iterative learning control without resetting condition for robot manipulators. *Robotica* 2011;29:975-80.
12. Chen Q, Lou Y. Compensated iterative learning control of industrial robots. 2018 IEEE International Conference on Real-time Computing and Robotics (RCAR); 2018. p. 52-7.
13. Ye Y, Tayebi A, Liu X. A unit-gain D-type iterative learning control scheme: application to a 6-dof robot manipulator. *IEEE* 2007.
14. Ratcliffe JD, Hätönen JJ, Lewin PL, Rogers E, Harte TJ, Owens DH. P-type iterative learning control for systems that contain resonance. *Int J Adapt Control Signal Process* 2005;19:769-96.
15. Dong J, He B, Zhang C, Li G. Open-Closed-Loop PD Iterative Learning Control with a Variable Forgetting Factor for a Two-Wheeled Self-Balancing Mobile Robot. *Complexity* 2019;2019:1-11.
16. Sun Y, Lin H, Li ZA. Open-loop PD-Type Iterative Learning Control for a Class of Nonlinear Systems with Control Delay and Arbitrary Initial Value [J]. Measurement and Control Technology, 31(6): 387-392, 2010. Available from: https://scholar.google.com.hk/scholar?hl=zh-CN&as_sdt=0%2C5&q=Open-loop+PD-Type+Iterative+Learning+Control+for+a+Class+of+Nonlinear+Systems+with+Control+Delay+and+Arbitrary+Initial+Value+%5BJ%5D.+Measurement+and+Control+Technology%2C+31%286%29%3A387-392%2C+2010&btnG [Last accessed on 18 Mar 2022]
17. Zhang L, Chen W, Liu J, Wen C. A robust adaptive iterative learning control for trajectory tracking of permanent-magnet spherical actuator. *IEEE Trans Ind Electron* 2016;63:291-301.
18. Boudjedir CE, Boukhetala D, Bouri M. Iterative learning control of a parallel delta robot. In: Chadli M, Bououden S, Ziani S, Zelinka I, editors. Advanced Control Engineering Methods in Electrical Engineering Systems. Cham: Springer International Publishing; 2019. p. 72-83.
19. Ma R, Zhang G. Iterative learning tracking control for a class of MIMO nonlinear time-varying systems. *International Journal of Modelling Identification and Control* 2017;27(4):271. Available from: https://www.inderscienceonline.com/doi/pdf/10.1504/IJMIC.2017.084721 [Last accessed on 18 Mar 2022]
20. Ouyang P, Zhang W, Gupta MM. An adaptive switching learning control method for trajectory tracking of robot manipulators. *Mechatronics* 2006;16:51-61.
21. Roveda L, Forgione M, Piga D. Robot control parameters auto-tuning in trajectory tracking applications. *Control Engineering Practice* 2020;101:104488.
22. Liu S, Meng D, Cheng L, et al. An iterative learning controller for a cable-driven hand rehabilitation robot. *IEEE* 2017.
23. Yang XF, Fan XP, Yang XY, et al. Open and closed loop PD-type iterative learning control of nonlinear system and its application in robots. *Journal of Changsha Railway Institute*, 2002;(01):78-84.
24. Wu G, Cui C, Wu G. dynamic modeling and torque feedforward based optimal fuzzy PD control of a high-speed parallel manipulator. *jrc* 2021;2.

AUTHOR INSTRUCTIONS

1. Submission Overview

Before you decide to publish with *Intelligence & Robotics* (*IR*), please read the following items carefully and make sure that you are well aware of Editorial Policies and the following requirements.

1.1 Topic Suitability

The topic of the manuscript must fit the scope of the journal. Please refer to Aims and Scope for more information.

1.2 Open Access and Copyright

The journal adopts Gold Open Access publishing model and distributes content under the Creative Commons Attribution 4.0 International License. Copyright is retained by authors. Please make sure that you are well aware of these policies.

1.3 Publication Fees

Before December 31, 2024, there are no article processing charges for papers accepted for publication after peer review. OAE subsidizes and helps authors publish their manuscripts totally free. For more details, please refer to OAE Publication Fees.

1.4 Language Editing

All submissions are required to be presented clearly and cohesively in good English. Authors whose first language is not English are advised to have their manuscripts checked or edited by a native English speaker before submission to ensure the high quality of expression. A well-organized manuscript in good English would make the peer review even the whole Editorial handling more smoothly and efficiently.

If needed, authors are recommended to consider the language editing services provided by Charlesworth to ensure that the manuscript is written in correct scientific English before submission. Authors who publish with OAE journals enjoy a special discount for the services of Charlesworth via the following two ways.

Submit your manuscripts directly at http://www.charlesworthauthorservices.com/~OAE;

Open the link http://www.charlesworthauthorservices.com/, and enter Promotion Code "OAE" when you submit.

1.5 Work Funded by the National Institutes of Health

If an accepted manuscript was funded by National Institutes of Health (NIH), the author may inform editors of the NIH funding number. The editors are able to deposit the paper to the NIH Manuscript Submission System on behalf of the author.

2. Submission Preparation

2.1 Cover Letter

A cover letter is required to be submitted accompanying each manuscript. Here is a guideline of a cover letter for authors' consideration:

List the highlights of the current manuscript and no more than 5 short sentences;

All authors have read the final manuscript, have approved the submission to the journal, and have accepted full responsibilities pertaining to the manuscript's delivery and contents;

Clearly state that the manuscript is an original work on its own merit, that it has not been previously published in whole or in part, and that it is not being considered for publication elsewhere;

No materials are reproduced from another source (if there is material in your manuscript that has been reproduced from another source, please state whether you have obtained permission from the copyright holder to use them);

Conflicts of interest statement;

If the manuscript is contributed to a Special Issue, please also mention it in the cover letter;

If the manuscript was presented partly or entirely in a conference, the author should clearly state the background information of the event, including the conference name, time, and place in the cover letter.

2.2 Types of Manuscripts

There is no restriction on the length of manuscripts, number of figures, tables and references, provided that the manuscript is concise and comprehensive. The journal publishes Research Article, Review, Technical Note, etc. For more details about paper type, please refer to the following table.

Manuscript Type	**Definition**	**Abstract**	**Keywords**	**Main Text Structure**
Research Article	A Research Article is a seminal and insightful research study and showcases that often involves modern techniques or methodologies. Authors should justify that their work is of novel findings.	The abstract should state briefly the purpose of the research, the principal results and major conclusions. No more than 250 words.	3-8 keywords	The main content should include four sections: Introduction, Methods, Results and Discussion.
Review	A Review should be an authoritative, well balanced, and critical survey of recent progress in an attractive or a fundamental research field.	Unstructured abstract. No more than 250 words.	3-8 keywords	The main text may consist of several sections with unfixed section titles. We suggest that the author include an "Introduction" section at the beginning, several sections with unfixed titles in the middle part, and a "Conclusions" section at the end.
Technical Note	A Technical Note is a short article giving a brief description of a specific development, technique, or procedure, or it may describe a modification of an existing technique, procedure or device applied in research.	Unstructured abstract. No more than 250 words.	3-8 keywords	/
Editorial	An Editorial is a short article describing news about the journal or opinions of senior Editors or the publisher.	None required	None required	/
Commentary	A Commentary is to provide comments on a newly published article or an alternative viewpoint on a certain topic.	Unstructured abstract. No more than 250 words.	3-8 keywords	/
Perspective	A Perspective provides personal points of view on the state-of-the-art of a specific area of knowledge and its future prospects.	Unstructured abstract. No more than 250 words.	3-8 keywords	/

2.3 Manuscript Structure

2.3.1 Front Matter

2.3.1.1 Title

The title of the manuscript should be concise, specific and relevant, with no more than 16 words if possible.

2.3.1.2 Authors and Affiliations

Authors' full names should be listed. The initials of middle names can be provided. The affiliations and email addresses for all authors should be listed. At least one author should be designated as the corresponding author. In addition, corresponding authors are suggested to provide their Open Researcher and Contributor ID upon submission. Please note that any change to authorship is not allowed after manuscript acceptance. The authors' affiliations should be provided in this format: department, institution, city, postcode, country.

2.3.1.3 Abstract

The abstract should be a single paragraph with word limitation and specific structure requirements (for more details please refer to Types of Manuscripts). It usually describes the main objective(s) of the study, explains how the study was done, including any model organisms used, without methodological detail, and summarizes the most important results and their significance. The abstract must be an objective representation of the study: it is not allowed to contain results that are not presented and substantiated in the manuscript, or exaggerate the main conclusions. Citations should not be included in the abstract.

2.3.1.4 Graphical Abstract

The graphical abstract is essential as this can catch first view of your publication by readers. We recommend you submit an eye-catching figure. It should summarize the content of the article in a concise graphical form. It is recommended to use it because this can make online articles get more attention.

The graphical abstract should be submitted as a separate document in the online submission system. Please provide an image with a minimum of 531 × 1328 pixels (h × w) or proportionally more. The image should be readable at a size of 5 cm × 13 cm using a regular screen resolution of 96 dpi. Preferred file types: TIFF, PSD, AI, JPEG, and EPS files.

2.3.1.5 Keywords

Three to eight keywords should be provided, which are specific to the article, yet reasonably common within the subject discipline.

2.3.2 Main Text

Manuscripts of different types are structured with different sections of content. Please refer to Types of Manuscripts to make sure which sections should be included in the manuscripts.

2.3.2.1 Introduction

The introduction should contain background that puts the manuscript into context, allow readers to understand why the study is important, include a brief review of key literature, and conclude with a brief statement of the overall aim of the work and a comment about whether that aim was achieved. Relevant controversies or disagreements in the field should be introduced as well.

2.3.2.2 Methods

The methods should contain sufficient details to allow others to fully replicate the study. New methods and protocols should be described in detail while well-established methods can be briefly described or appropriately cited. Statistical terms, abbreviations, and all symbols used should be defined clearly. Protocol documents for clinical trials, observational studies, and other non-laboratory investigations may be uploaded as supplementary materials.

2.3.2.3 Results

This section contains the findings of the study. Results of statistical analysis should also be included either as text or as tables or figures if appropriate. Authors should emphasize and summarize only the most important observations. Data on all primary and secondary outcomes identified in the section Methods should also be provided. Extra or supplementary materials and technical details can be placed in supplementary documents.

2.3.2.4 Discussion

This section should discuss the implications of the findings in context of existing research and highlight limitations of the study. Future research directions may also be mentioned.

2.3.2.5 Conclusion

It should state clearly the main conclusions and include the explanation of their relevance or importance to the field.

2.3.3 Back Matter

2.3.3.1 Acknowledgments

Anyone who contributed towards the article but does not meet the criteria for authorship, including those who provided professional writing services or materials, should be acknowledged. Authors should obtain permission to acknowledge from all those mentioned in the Acknowledgments section. This section is not added if the author does not have anyone to acknowledge.

2.3.3.2 Authors' Contributions

Each author is expected to have made substantial contributions to the conception or design of the work, or the acquisition, analysis, or interpretation of data, or the creation of new software used in the work, or have drafted the work or substantively revised it.

Please use Surname and Initial of Forename to refer to an author's contribution. For example: made substantial contributions to conception and design of the study and performed data analysis and interpretation: Salas H, Castaneda WV; performed data acquisition, as well as providing administrative, technical, and material support: Castillo N, Young V.

If an article is single-authored, please include "The author contributed solely to the article." in this section.

2.3.3.3 Availability of Data and Materials

In order to maintain the integrity, transparency and reproducibility of research records, authors should include this section in their manuscripts, detailing where the data supporting their findings can be found. Data can be deposited into data repositories or published as supplementary information in the journal. Authors who cannot share their data should state that the data will not be shared and explain it. If a manuscript does not involve such issues, please state "Not applicable." in this section.

2.3.3.4 Financial Support and Sponsorship

All sources of funding for the study reported should be declared. The role of the funding body in the experiment design, collection, analysis and interpretation of data, and writing of the manuscript should be declared. Any relevant grant numbers and the link of funder's website should be provided if any. If the study is not involved with this issue, state "None." in this section.

2.3.3.5 Conflicts of Interest

Authors must declare any potential conflicts of interest that may be perceived as inappropriately influencing the representation or interpretation of reported research results. If there are no conflicts of interest, please state “All authors declared that there are no conflicts of interest.” in this section. Some authors may be bound by confidentiality agreements. In such cases, in place of itemized disclosures, we will require authors to state “All authors declared that they are bound by confidentiality agreements that prevent them from disclosing their conflicts of interest in this work.”. If authors are unsure whether conflicts of interest exist, please refer to the “Conflicts of Interest” of *IR* Editorial Policies for a full explanation.

2.3.3.6 Ethical Approval and Consent to Participate

Research involving human subjects, human material or human data must be performed in accordance with the Declaration of Helsinki and approved by an appropriate ethics committee. An informed consent to participate in the study should also be obtained from participants, or their parents or legal guardians for children under 16. A statement detailing the name of the ethics committee (including the reference number where appropriate) and the informed consent obtained must appear in the manuscripts reporting such research.

Studies involving animals and cell lines must include a statement on ethical approval. More information is available at Editorial Policies.

If the manuscript does not involve such issue, please state “Not applicable.” in this section.

2.3.3.7 Consent for Publication

Manuscripts containing individual details, images or videos, must obtain consent for publication from that person, or in the case of children, their parents or legal guardians. If the person has died, consent for publication must be obtained from the next of kin of the participant. Manuscripts must include a statement that written informed consent for publication was obtained. Authors do not have to submit such content accompanying the manuscript. However, these documents must be available if requested. If the manuscript does not involve this issue, state “Not applicable.” in this section.

2.3.3.8 Copyright

Authors retain copyright of their works through a Creative Commons Attribution 4.0 International License that clearly states how readers can copy, distribute, and use their attributed research, free of charge. A declaration “© The Author(s) 2022.” will be added to each article. Authors are required to sign License to Publish before formal publication.

2.3.3.9 References

References should be numbered in order of appearance at the end of manuscripts. In the text, reference numbers should be placed in square brackets and the corresponding references are cited thereafter. If the number of authors is less than or equal to six, we require to list all authors’ names. If the number of authors is more than six, only the first three authors’ names are required to be listed in the references, other authors’ names should be omitted and replaced with “et al.”. Abbreviations of the journals should be provided on the basis of Index Medicus. Information from manuscripts accepted but not published should be cited in the text as “Unpublished material” with written permission from the source.

References should be described as follows, depending on the types of works:

Types	**Examples**
Journal articles by individual authors	Weaver DL, Ashikaga T, Krag DN, et al. Effect of occult metastases on survival in node-negative breast cancer. *N Engl J Med* 2011;364:412-21. [PMID: 21247310 DOI: 10.1056/NEJMoa1008108]
Organization as author	Diabetes Prevention Program Research Group. Hypertension, insulin, and proinsulin in participants with impaired glucose tolerance. *Hypertension* 2002;40:679-86. [DOI: 10.1161/01.HYP.0000035706.28494.09]
Both personal authors and organization as author	Vallancien G, Emberton M, Harving N, van Moorselaar RJ; Alf-One Study Group. Sexual dysfunction in 1,274 European men suffering from lower urinary tract symptoms. *J Urol* 2003;169:2257-61. [PMID: 12771764 DOI: 10.1097/01.ju.0000067940.76090.73]
Journal articles not in English	Zhang X, Xiong H, Ji TY, Zhang YH, Wang Y. Case report of anti-N-methyl-D-aspartate receptor encephalitis in child. *J Appl Clin Pediatr* 2012;27:1903-7. (in Chinese)
Journal articles ahead of print	Odibo AO. Falling stillbirth and neonatal mortality rates in twin gestation: not a reason for complacency. BJOG 2018; Epub ahead of print [PMID: 30461178 DOI: 10.1111/1471-0528.15541]
Books	Sherlock S, Dooley J. Diseases of the liver and billiary system. 9th ed. Oxford: Blackwell Sci Pub; 1993. pp. 258-96.
Book chapters	Meltzer PS, Kallioniemi A, Trent JM. Chromosome alterations in human solid tumors. In: Vogelstein B, Kinzler KW, editors. The genetic basis of human cancer. New York: McGraw-Hill; 2002. pp. 93-113.
Online resource	FDA News Release. FDA approval brings first gene therapy to the United States. Available from: https://www.fda.gov/NewsEvents/Newsroom/PressAnnouncements/ucm574058.htm. [Last accessed on 30 Oct 2017]

Conference proceedings	Harnden P, Joffe JK, Jones WG, Editors. Germ cell tumours V. Proceedings of the 5th Germ Cell Tumour Conference; 2001 Sep 13-15; Leeds, UK. New York: Springer; 2002.
Conference paper	Christensen S, Oppacher F. An analysis of Koza's computational effort statistic for genetic programming. In: Foster JA, Lutton E, Miller J, Ryan C, Tettamanzi AG, editors. Genetic programming. EuroGP 2002: Proceedings of the 5th European Conference on Genetic Programming; 2002 Apr 3-5; Kinsdale, Ireland. Berlin: Springer; 2002. pp. 182-91.
Unpublished material	Tian D, Araki H, Stahl E, Bergelson J, Kreitman M. Signature of balancing selection in Arabidopsis. *Proc Natl Acad Sci U S A*. Forthcoming 2002.

The journal also recommends that authors prepare references with a bibliography software package, such as EndNote to avoid typing mistakes and duplicated references.

2.3.3.10 Supplementary Materials

Additional data and information can be uploaded as Supplementary Materials to accompany the manuscripts. The supplementary materials will also be available to the referees as part of the peer-review process. Any file format is acceptable, such as data sheet (word, excel, csv, cdx, fasta, pdf or zip files), presentation (powerpoint, pdf or zip files), image (cdx, eps, jpeg, pdf, png or tiff), table (word, excel, csv or pdf), audio (mp3, wav or wma) or video (avi, divx, flv, mov, mp4, mpeg, mpg or wmv). All information should be clearly presented. Supplementary materials should be cited in the main text in numeric order (e.g., Supplementary Figure 1, Supplementary Figure 2, Supplementary Table 1, Supplementary Table 2, *etc*.). The style of supplementary figures or tables complies with the same requirements on figures or tables in main text. Videos and audios should be prepared in English, and limited to a size of 500 MB.

2.4 Manuscript Format

2.4.1 File Format

Manuscript files can be in DOC and DOCX formats and should not be locked or protected.

Manuscript prepared in LaTex must be collated into one ZIP folder (including all source files and images, so that the Editorial Office can recompile the submitted PDF).

When preparing manuscripts in different file formats, please use the corresponding Manuscript Templates.

2.4.2 Length

There are no restrictions on paper length, number of figures, or number of supporting documents. Authors are encouraged to present and discuss their findings concisely.

2.4.3 Language

Manuscripts must be written in English.

2.4.4 Multimedia Files

The journal supports manuscripts with multimedia files. The requirements are listed as follows:

Video or audio files are only acceptable in English. The presentation and introduction should be easy to understand. The frames should be clear, and the speech speed should be moderate;

A brief overview of the video or audio files should be given in the manuscript text;

The video or audio files should be limited to a size of up to 500 MB;

Please use professional software to produce high-quality video files, to facilitate acceptance and publication along with the submitted article. Upload the videos in mp4, wmv, or rm format (preferably mp4) and audio files in mp3 or wav format.

2.4.5 Figures

Figures should be cited in numeric order (e.g., Figure 1, Figure 2) and placed after the paragraph where it is first cited;

Figures can be submitted in format of TIFF, PSD, AI, EPS or JPEG, with resolution of 300-600 dpi;

Figure caption is placed under the Figure;

Diagrams with describing words (including, flow chart, coordinate diagram, bar chart, line chart, and scatter diagram, *etc*.) should be editable in word, excel or powerpoint format. Non-English information should be avoided;

Labels, numbers, letters, arrows, and symbols in figure should be clear, of uniform size, and contrast with the background;

Symbols, arrows, numbers, or letters used to identify parts of the illustrations must be identified and explained in the legend;

Internal scale (magnification) should be explained and the staining method in photomicrographs should be identified;

All non-standard abbreviations should be explained in the legend;

Permission for use of copyrighted materials from other sources, including re-published, adapted, modified, or partial figures and images from the internet, must be obtained. It is authors' responsibility to acquire the licenses, to follow any citation instruction requested by third-party rights holders, and cover any supplementary charges.

2.4.6 Tables

Tables should be cited in numeric order and placed after the paragraph where it is first cited;
The table caption should be placed above the table and labeled sequentially (e.g., Table 1, Table 2);
Tables should be provided in editable form like DOC or DOCX format (picture is not allowed);
Abbreviations and symbols used in table should be explained in footnote;
Explanatory matter should also be placed in footnotes;
Permission for use of copyrighted materials from other sources, including re-published, adapted, modified, or partial tables from the internet, must be obtained. It is authors' responsibility to acquire the licenses, to follow any citation instruction requested by third-party rights holders, and cover any supplementary charges.

2.4.7 Abbreviations

Abbreviations should be defined upon first appearance in the abstract, main text, and in figure or table captions and used consistently thereafter. Non-standard abbreviations are not allowed unless they appear at least three times in the text. Commonly-used abbreviations, such as DNA, RNA, ATP, *etc*., can be used directly without definition. Abbreviations in titles and keywords should be avoided, except for the ones which are widely used.

2.4.8 Italics

General italic words like *vs*., *et al*., *etc*., *in vivo*, *in vitro*; *t* test, *F* test, *U* test; related coefficient as *r*, sample number as *n*, and probability as *P*; names of genes; names of bacteria and biology species in Latin.

2.4.9 Units

SI Units should be used. Imperial, US customary and other units should be converted to SI units whenever possible. There is a space between the number and the unit (i.e., 23 mL). Hour, minute, second should be written as h, min, s.

2.4.10 Numbers

Numbers appearing at the beginning of sentences should be expressed in English. When there are two or more numbers in a paragraph, they should be expressed as Arabic numerals; when there is only one number in a paragraph, number < 10 should be expressed in English and number > 10 should be expressed as Arabic numerals. 12345678 should be written as 12,345,678.

2.4.11 Equations

Equations should be editable and not appear in a picture format. Authors are advised to use either the Microsoft Equation Editor or the MathType for display and inline equations.
Display equations should be numbered consecutively, using Arabic numbers in parentheses;
Inline equations should not be numbered, with the same/similar size font used for the main text.

2.4.12 Headings

In the main body of the paper, three different levels of headings may be used.
Level one headings: they should be in bold, and numbered using Arabic numbers, such as **1. INTRODUCTION**, and **2. METHODS**, with all letters capitalized;
Level two headings: they should be in bold and numbered after the level one heading, such as **2.1 Statistical analyses**, **2.2** ..., **2.3**..., *etc*., with the first letter capitalized;
Level three headings: they should be italicized, and numbered after the level two heading, such as *2.1.1 Data distributions*,and *2.1.2 outliers and linear regression*, with the first letter capitalized.

2.4.13 Text Layout

As the electronic submission will provide the basic material for typesetting, it is important to prepare papers in the general editorial style of the journal.
The font is Times New Roman;
The font size is 12pt;
Single column, 1.5× line spacing;
Insert one line break (one Return) before the heading and paragraph, if the heading and paragraph are adjacent, insert a line break before the heading only;
No special indentation;
Alignment is left end;
Insert consecutive line numbers;
For other details please refer to the Manuscript Templates.

2.5 Submission Link

Submit an article via https://oaemesas.com/login?JournalId=ir.

3. Publication Ethics Statement

OAE is a member of the Committee on Publication Ethics (COPE). We fully adhere to its Code of Conduct and to its Best Practice Guidelines.

The Editors of this journal enforce a rigorous peer-review process together with strict ethical policies and standards to guarantee to add high-quality scientific works to the field of scholarly publication. Unfortunately, cases of plagiarism, data falsification, image manipulation, inappropriate authorship credit, and the like, do arise. The Editors of *IR* take such publishing ethics issues very seriously and are trained to proceed in such cases with zero tolerance policy.

Authors wishing to publish their papers in *IR* must abide by the following:

The author(s) must disclose any possibility of a conflict of interest in the paper prior to submission;
The authors should declare that there is no academic misconduct in their manuscript in the cover letter;
Authors should accurately present their research findings and include an objective discussion of the significance of their findings;
Data and methods used in the research need to be presented in sufficient detail in the manuscript so that other researchers can replicate the work;
Authors should provide raw data if referees and the Editors of the journal request;
Simultaneous submission of manuscripts to more than one journal is not tolerated;
Republishing content that is not novel is not tolerated (for example, an English translation of a paper that is already published in another language will not be accepted);
The manuscript should not contain any information that has already been published. If you include already published figures or images, please get the necessary permission from the copyright holder to publish under the CC-BY license;
Plagiarism, data fabrication and image manipulation are not tolerated;
Plagiarism is not acceptable in OAE journals.

Plagiarism involves the inclusion of large sections of unaltered or minimally altered text from an existing source without appropriate and unambiguous attribution, and/or an attempt to misattribute original authorship regarding ideas or results, and copying text, images, or data from another source, even from your own publications, without giving credit to the source.

As to reusing the text that is copied from another source, it must be between quotation marks and the source must be cited. If a study's design or the manuscript's structure or language has been inspired by previous studies, these studies must be cited explicitly.

If plagiarism is detected during the peer-review process, the manuscript may be rejected. If plagiarism is detected after publication, we may publish a Correction or retract the paper.

Falsification is manipulating research materials, equipment, or processes, or changing or omitting data or results so that the findings are not accurately represented in the research record.

Image files must not be manipulated or adjusted in any way that could lead to misinterpretation of the information provided by the original image.

Irregular manipulation includes: introduction, enhancement, moving, or removing features from the original image; the grouping of images that should be presented separately, or modifying the contrast, brightness, or color balance to obscure, eliminate, or enhance some information.

If irregular image manipulation is identified and confirmed during the peer-review process, we may reject the manuscript. If irregular image manipulation is identified and confirmed after publication, we may publish a Correction or retract the paper.

OAE reserves the right to contact the authors' institution(s) to investigate possible publication misconduct if the Editors find conclusive evidence of misconduct before or after publication. OAE has a partnership with iThenticate, which is the most trusted similarity checker. It is used to analyze received manuscripts to avoid plagiarism to the greatest extent possible. When plagiarism becomes evident after publication, we will retract the original publication or require modifications, depending on the degree of plagiarism, context within the published article, and its impact on the overall integrity of the published study. Journal Editors will act under the relevant COPE guidelines.

4. Authorship

Authorship credit of *IR* should be solely based on substantial contributions to a published study, as specified in the following four criteria:

1. Substantial contributions to the conception or design of the work, or the acquisition, analysis, or interpretation of data for the work;
2. Drafting the work or revising it critically for important intellectual content;
3. Final approval of the version to be published;
4. Agreement to be accountable for all aspects of the work in ensuring that questions related to the accuracy or integrity of any part of the work are appropriately investigated and resolved.

All those who meet these criteria should be identified as authors. Authors must specify their contributions in the section Authors' Contributions of their manuscripts. Contributors who do not meet all the four criteria (like only involved in acquisition of funding, general supervision of a research group, general administrative support, writing assistance, technical editing, language editing, proofreading, *etc*.) should be acknowledged in the section of Acknowledgement in the manuscript rather than being listed as authors.

If a large multiple-author group has conducted the work, the group ideally should decide who will be authors before the work starts and confirm authors before submission. All authors of the group named as authors must meet all the four criteria for authorship.

5. Reviewers Exclusions

You are welcome to exclude a limited number of researchers as potential Editors or reviewers of your manuscript. To ensure a fair and rigorous peer review process, we ask that you keep your exclusions to a maximum of three people. If you wish to exclude additional referees, please explain or justify your concerns—this information will be helpful for Editors when deciding whether to honor your request.

6. Editors and Journal Staff as Authors

Editorial independence is extremely important and OAE does not interfere with Editorial decisions. Editorial staff or Editors shall not be involved in processing their own academic work. Submissions authored by Editorial staff/Editors will be assigned to at least two independent outside reviewers. Decisions will be made by the Editor-in-Chief, including Special Issue papers. Journal staff are not involved in the processing of their own work submitted to any OAE journals.

7. Conflict of Interests

OAE journals require authors to declare any possible financial and/or non-financial conflicts of interest at the end of their manuscript and in the cover letter, as well as confirm this point when submitting their manuscript in the submission system. If no conflicts of interest exist, authors need to state "All authors declared that there are no conflicts of interest". We also recognize that some authors may be bound by confidentiality agreements, in which cases authors need to state "All authors declared that they are bound by confidentiality agreements that prevent them from disclosing their competing interests in this work".

8. Editorial Process

8.1. Pre-Check

New submissions are initially checked by the Managing Editor from the perspectives of originality, suitability, structure and formatting, conflicts of interest, background of authors, *etc*. Poorly prepared manuscripts may be rejected at this stage. If your manuscript does not meet one or more of these requirements, we will return it for further revisions.

Once your manuscript has passed the initial check, it will be assigned to the Assistant Editor, and then the Editor-in-Chief, or an Associate Editor in the case of a conflict of interest, will be notified of the submission and invited to review. Regarding Special Issue paper, after passing the initial check, the manuscript will be successively assigned to the Assistant Editor, and then to the Editor-in-Chief, or an Associate Editor in the case of conflict of interest for the Editor-in-Chief to review. The Editor-in-Chief, or the Associate Editor may reject manuscripts that they deem highly unlikely to pass peer review without further consultation. Once your manuscript has passed the Editorial assessment, the Associate Editor will start to organize peer-review.

All manuscripts submitted to *IR* are screened using CrossCheck powered by iThenticate to identify any plagiarized content. Your study must also meet all ethical requirements as outlined in our Editorial Policies. If the manuscript does not pass any of these checks, we may return it to you for further revisions or decline to consider your study for publication.

8.2. Peer Review

IR operates a single-blind review process, which means that reviewers know the names of authors, but the names of the reviewers are hidden from the authors. The scientific quality of the research described in the manuscript is assessed by a minimum of two independent expert reviewers. The Editor-in-Chief is responsible for the final decision regarding acceptance or rejection of the manuscript.

All information contained in your manuscript and acquired during the review process will be held in the strictest confidence.

8.3. Decisions

Your research will be judged on scientific soundness only, not on its perceived impact as judged by Editors or referees. There are three possible decisions: Accept (your study satisfies all publication criteria), Invitation to Revise (more work is required to satisfy all criteria), and Reject (your study fails to satisfy key criteria and it is highly unlikely that further work can address its shortcomings). All of the following publication criteria must be fulfilled to enable your manuscript to be accepted for publication:

Originality
The study reports original research and conclusions.
Data availability
All data to support the conclusions either have been provided or are otherwise publicly available.
Statistics
All data have been analyzed through appropriate statistical tests and these are clearly defined.
Methods
The methods are described in sufficient detail to be replicated.
Citations
Previous work has been appropriately acknowledged.
Interpretation
The conclusions are a reasonable extension of the results.
Ethics
The study design, data presentation, and writing style comply with our Editorial Policies.

8.4. Revisions

Authors are required to submit the revised manuscript within one week if minor revision is recommended while two weeks if major revision recommended or one month if additional experiments are needed. If authors need more than one month to revise their manuscript, we usually require the authors to resubmit their paper. We request that a document of point-to-point response to all comments of reviewers and the Editor-in-Chief or the Associate Editor should be supplied along with the revised manuscript to allow quick assessment of your revised manuscript. This document should outline in detail how each of the comments was addressed in the revised manuscript or should provide a rebuttal to the criticism. Manuscripts may or may not be sent to reviewers after revision, dependent on whether the reviewer requested to see the revised version. Apart from in exceptional circumstances, *IR* only supports a round of major revision per manuscript.

9. Contact Us

Journal Contact

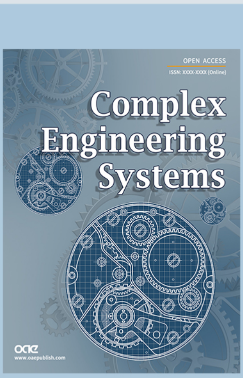
Intelligence & Robotics Editorial Office
Suite 1504, Plaza A, Xi’an National Digital Publishing Base,
No. 996 Tiangu 7th Road, Gaoxin District, Xi’an 710077, Shaanxi, China.

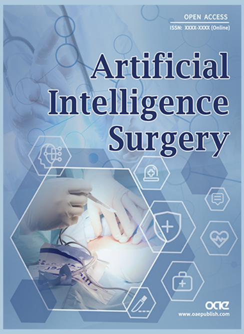
Managing Editor

Lijun Jin
Email: editorial@intellrobot.com

OAE Publishing Inc. (https://oaepublish.com/) is a multidisciplinary open-access publishing company, founded in Los Angeles in 2015. Until now, OAE has been recognized by authoritative organizations in publishing industries, such as the ORCID, COPE, Scientific, Technical and Medical Publishers (STM), Crossref, and EASE.

As of July 2021, more than 1,200 outstanding scholars have joined OAE, who are from world-renowned universities and research institutions, including European Academy of Sciences, American Academy of Invention Sciences, Chinese Academy of Sciences, Royal Academy of Sciences of Belgium, British Academy of Medical Sciences, etc. There are more than 30 journals founded by OAE (https://oaepublish.com/about/journals), such as Intelligence & Robotics, Journal of Materials Informatics, Complex Engineering Systems, Journal of Smart Environments and Green Computing, and Soft Science, etc. Part of journals have been indexed by Scopus and CAS. We are currently working on database application including PubMed and ESCI. Up to July 2021, 2,354 articles have been published online, with 13,131,129 hits and 963,586 downloads. In the future, OAE Publishing Company will continue to found more quality journals with outstanding scholars, to promote the global academic development.





OAE Official Website