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

Intelligence & Robotics

Research Article

Open Access

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

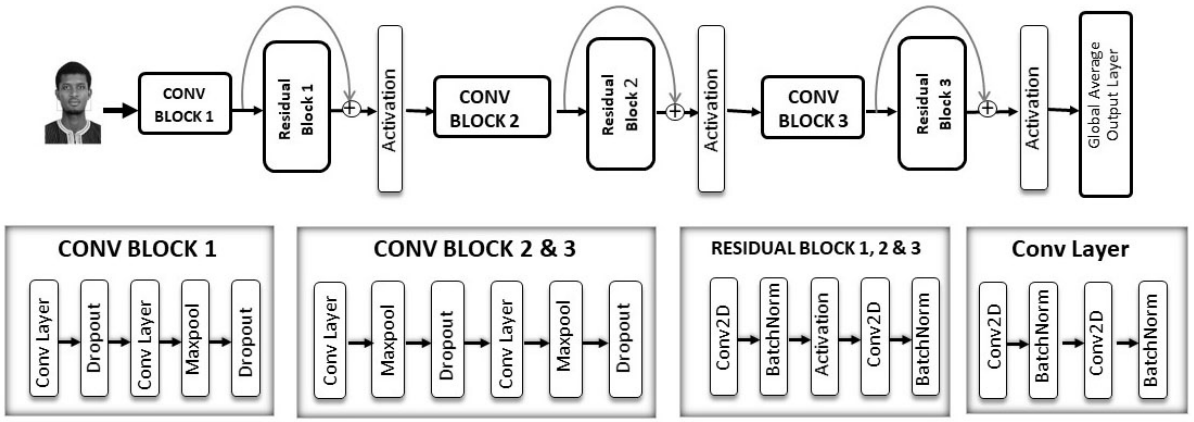

Figure 1. Architecture model of the proposed framework.

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:

$$OS = \frac{IS - KS + 2P}{S} + 1 \quad (1)$$

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

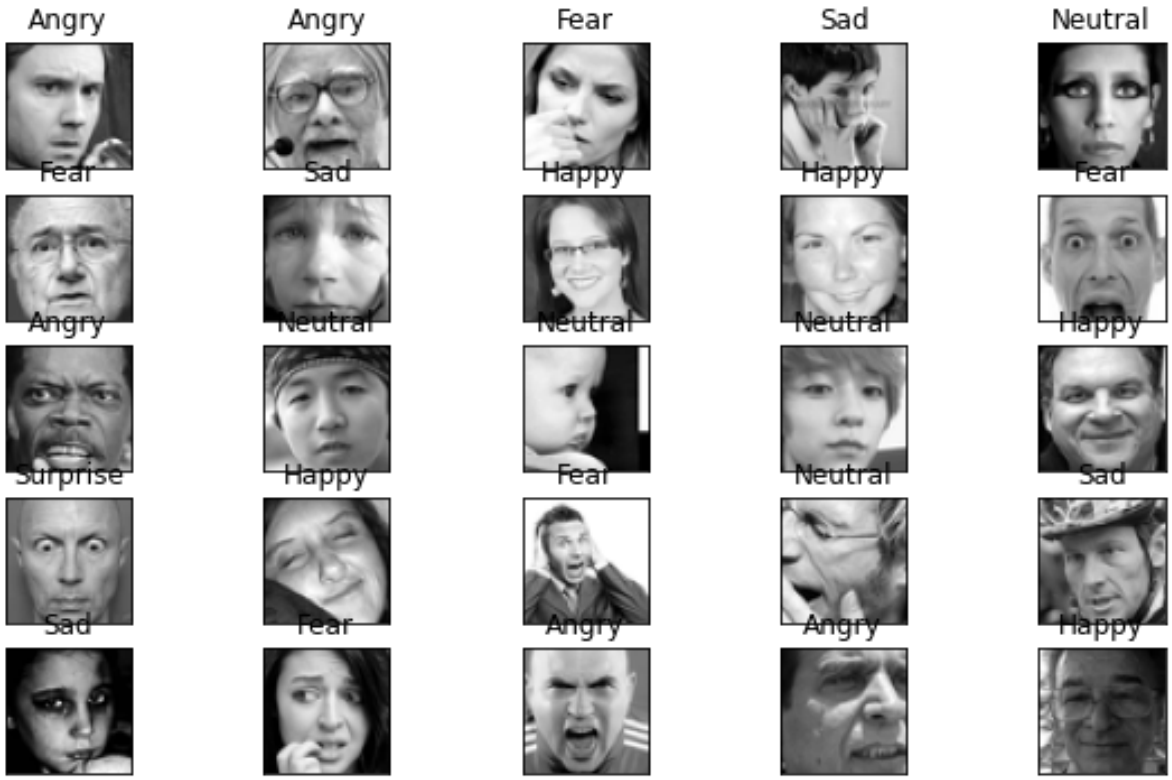

Figure 2. The seven expressions included in the FER-2013 dataset (anger, fear, happiness, sadness, surprise, disgust, and neutral).

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

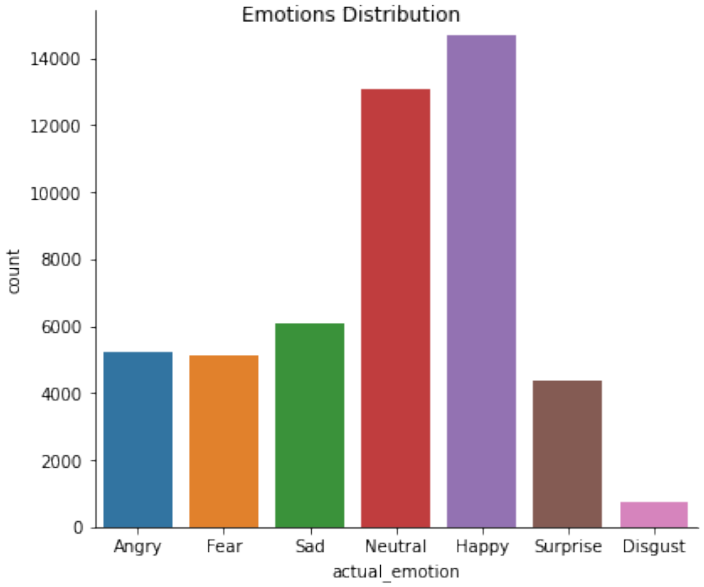

Figure 3. Data distribution among the 7 emotions.

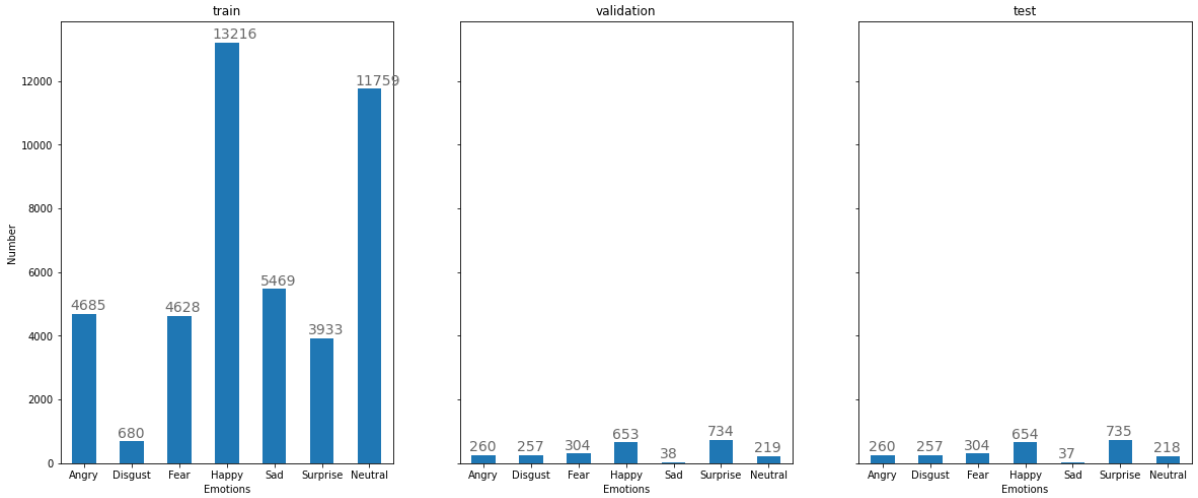

Figure 4. Data split into training, validation and testing sets.

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.

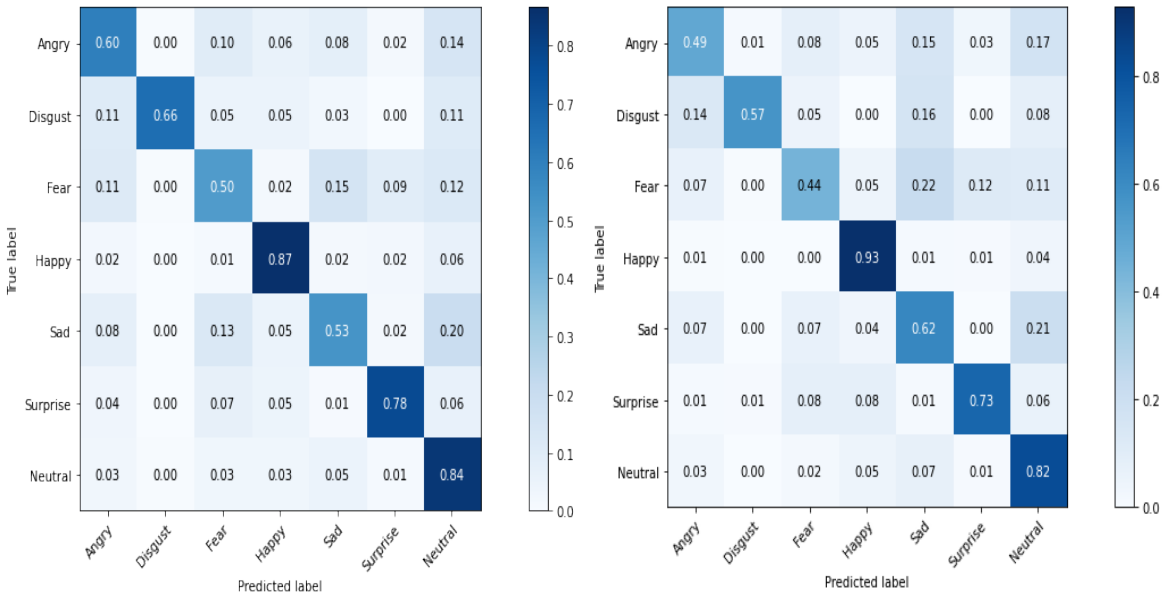

Figure 5. Confusion matrix for the FER showing the prediction accuracy of the emotional expressions using the proposed architecture with residual blocks (left) and the basic architecture (right).

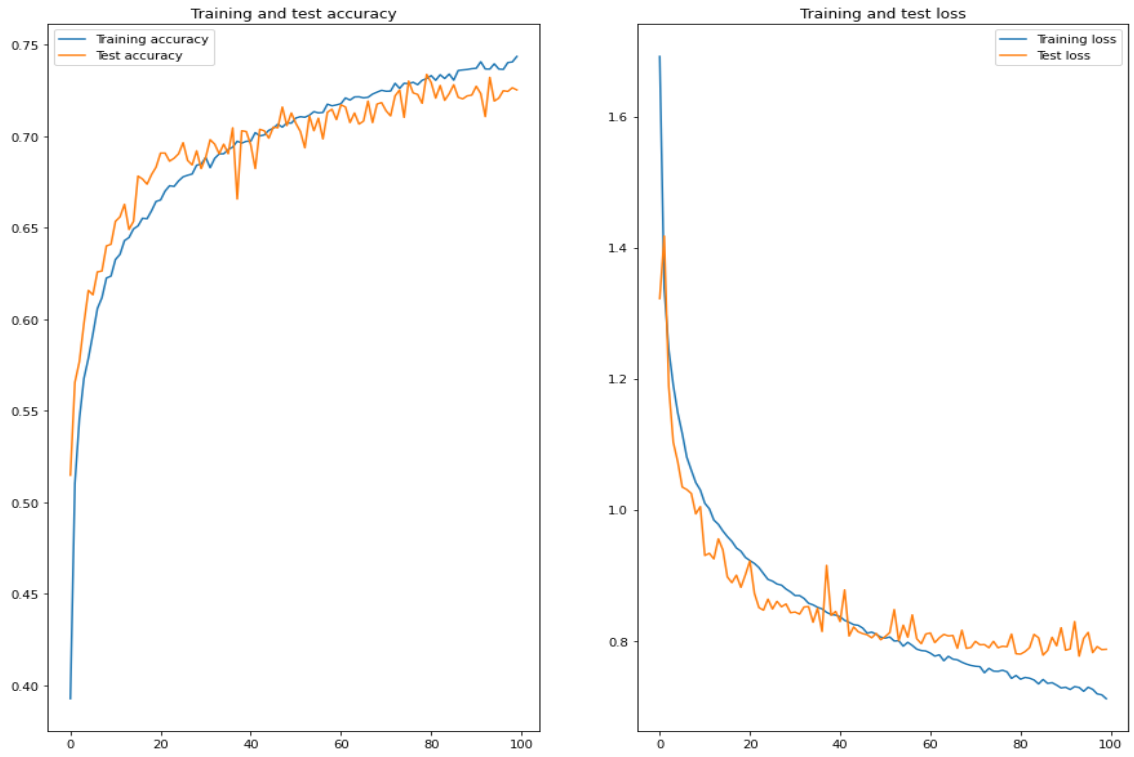

Figure 6. The accuracy and the loss of the training and validation data of FERGIT on the basic model over 100 epochs.

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

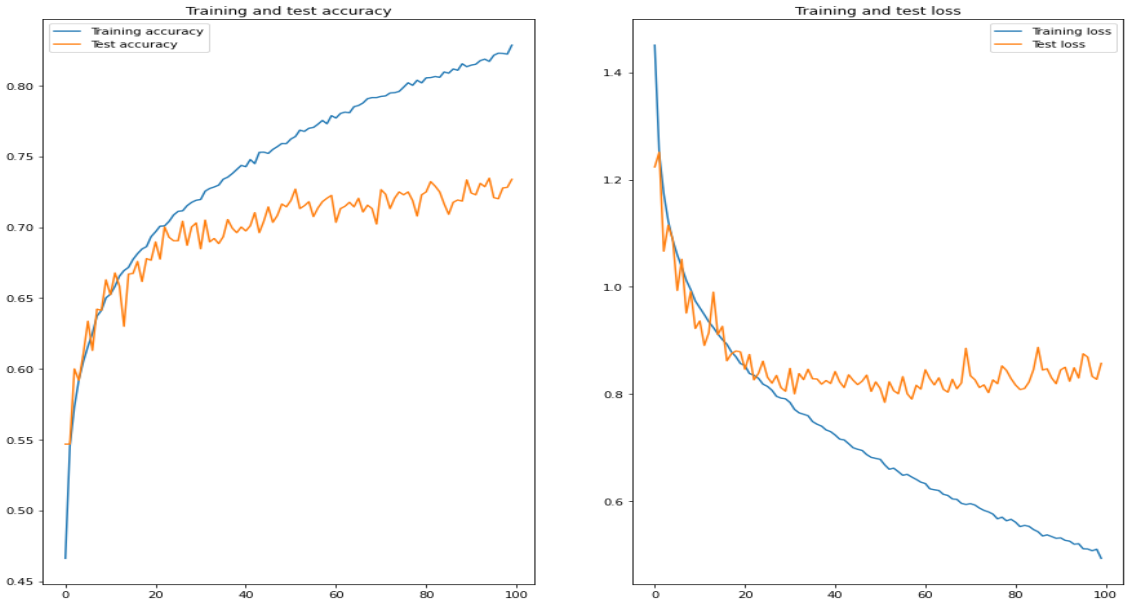

Figure 7. The accuracy and the loss of the training and validation data of FERGIT on the ResNet based model over 100 epochs.

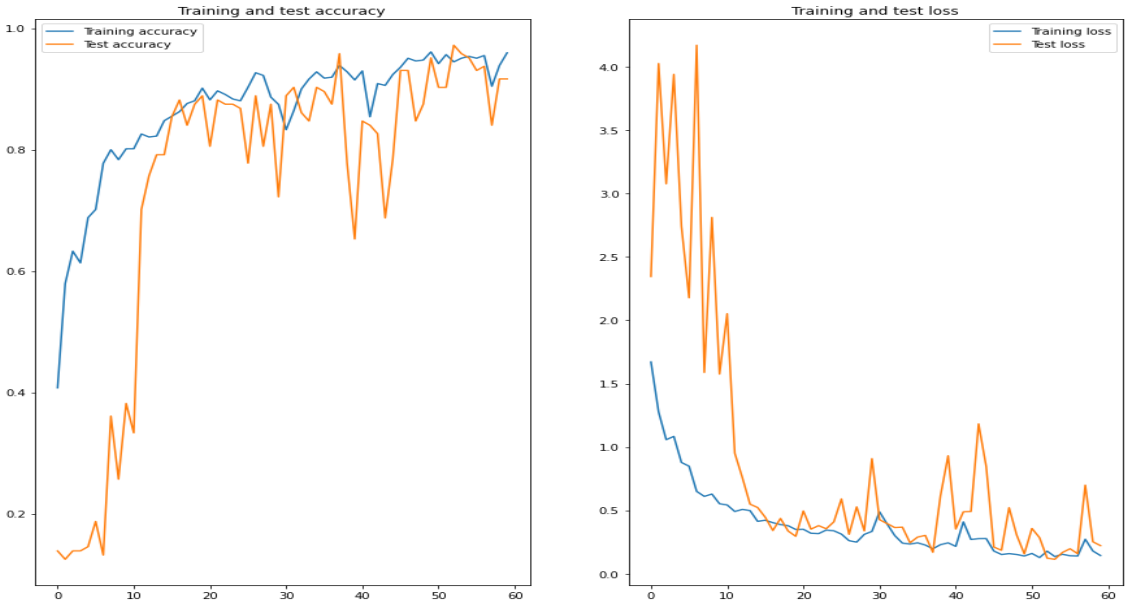

Figure 8. The accuracy and the loss of the training and validation data of CK+ on the basic model over 100 epochs.

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**

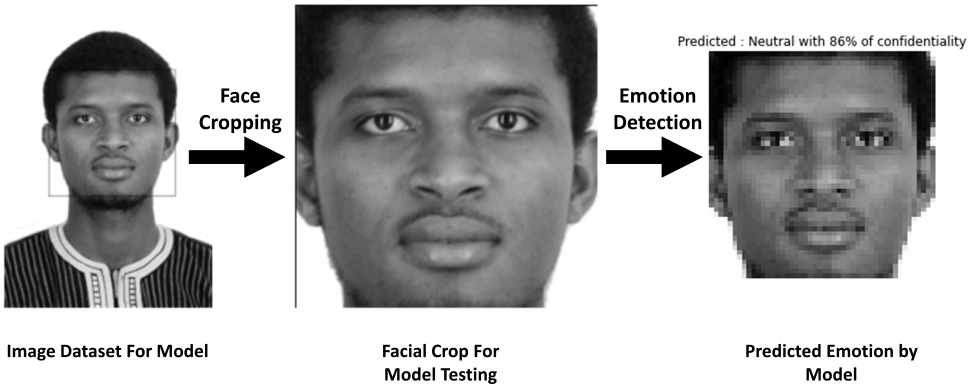

Figure 9. Framework testing on an individual pose

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.

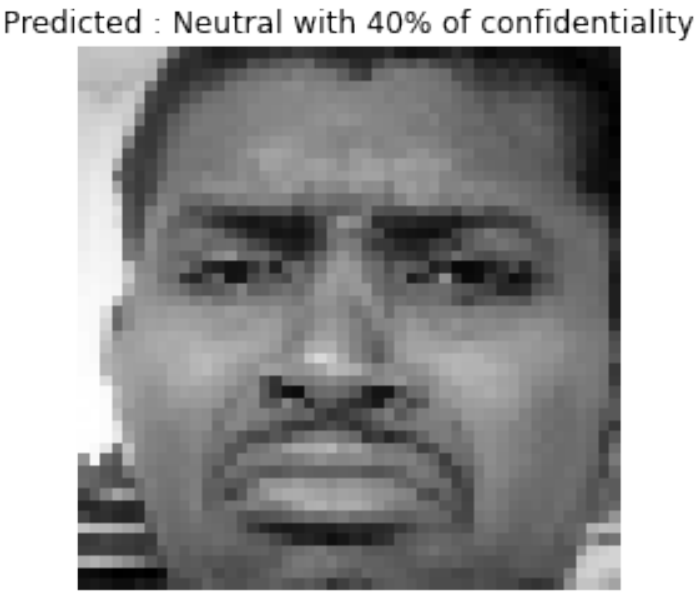

Figure 10. Wrong prediction on the individual facial image.

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

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.

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