Nexo Revista Científica
Vol. 36, No. 05(Especial),pp. 01-18/Noviembre 2023

ISSN-E 1995-9516
Universidad Nacional de Ingeniería

http://revistas.uni.edu.ni/index.php/Nexo
https://doi.org/10.5377/nexo.v36i05.17181

Deep Learning Approaches for Classification of Emotion Recognition based on Facial Expressions

Enfoques de aprendizaje profundo para la clasificación del reconocimiento de emociones basado en expresiones faciales

Ahmed Adnan Hameed Qutub[1,*] and Yılmaz Atay [2]
[1] Graduate School of Natural and Applied Sciences, Gazi University, Ankara, Türkiye.
[2] Department of Computer Engineering, Faculty of Engineering, Gazi University, Ankara, Türkiye.

[*] aadnan.qutub@gazi.edu.tr /ORCID / https://orcid.org/ 0009-0007-9346-2501
yilmazatay@gazi.edu.tr / ORCID / https://orcid.org/0000-0002-3298-3334

(received: 18-september-2023; /accepted: November 30, 2023)

ABSTRACT

Automated emotion recognition is crucial in numerous industries that depend on understanding human emotional responses, such as advertising, technology, and human–robot interaction, particularly within the information technology field. However, current systems are often insufficient for comprehensively understanding an individual's emotions, as previous research has mainly focused on assessing facial expressions and categorizing them into seven primary emotions, including neutrality. In this study, we present several deep convolutional neural network (CNN) models explicitly designed for facial emotion recognition, using the FER2013 and RAF datasets. The baseline CNN model is established by trial-and-error, and its results are compared with more complex deep learning techniques, including ResNet18, VGGNet16, VGGNet19, and EfficientNet-B0 models. Among these, the VGGNet19 model achieved the best results with a test accuracy of 71.02% on the FER2013 dataset. The ResNet18 model outperformed all other models with an 86.02% test accuracy on the RAF-DB dataset. These results will contribute to potential studies on automatic emotion recognition and classification with different deep-learning techniques.

Keywords: Gesture recognition, deep learning models, VGGNet, ResNet, EfficientNet.

RESUMEN

El reconocimiento automático de emociones juega un papel crucial en numerosas industrias que dependen de comprender las respuestas emocionales humanas, como la publicidad, la tecnología y la interacción humano-robot, especialmente dentro del campo de la Tecnología de la Información (TI). Sin embargo, los sistemas actuales a menudo no logran comprender de manera integral las emociones de un individuo, ya que las investigaciones previas se han centrado principalmente en evaluar las expresiones faciales y clasificarlas en siete emociones primarias, incluida la neutralidad. En este estudio, presentamos varios modelos de Redes Neuronales Convolucionales Profundas (CNN) diseñados específicamente para la tarea de reconocimiento facial de emociones, utilizando los conjuntos de datos FER2013 y RAF. El modelo base de CNN se establece mediante un método de prueba y error, y sus resultados se comparan con técnicas de aprendizaje profundo más complejas, que incluyen los modelos ResNet18, VGGNet16, VGGNet19 y EfficientNet-B0. Entre estos modelos, el modelo VGGNet19 logró los mejores resultados con una precisión de prueba del 71.02% en el conjunto de datos FER2013. En comparación, el modelo ResNet18 superó a todos los demás modelos con una precisión de prueba del 86.02% en el conjunto de datos RAF-DB. Estos resultados destacan el potencial para avanzar en el reconocimiento automático de emociones a través de técnicas complejas de aprendizaje profundo.

Palabras clave: Reconocimiento de gestos, modelos de aprendizaje profundo, VGGNet, ResNet, EfficientNet.

1. INTRODUCTION

Artificial intelligence and computer vision researchers have focused on the development of automatic facial expression recognition (FER) for the decade. Researchers have achieved remarkable results in identifying emotions from facial expressions (eg, Bartlett, Stewart, et al., 2004). Furthermore, movement and gesture have been explored as essential modes of nonverbal communication in human–human and human–computer interactions (Hudlicka, Eva, 2003). However, most studies of automatic emotion recognition have concentrated on the face; only recently some researchers have begun to incorporate gestures to express emotions in human–computer interactions.

With technological advancements, automated emotion recognition (AEE) has become a pivotal component within industries dependent on human emotional responses. AEE encompasses fields such as advertising, technology, and human–robot interaction, where it maintains a specific emphasis and holds significant importance, especially within the information technology sector. However, the challenges posed by cultural and gender differences make AEE more difficult (Allen & Pease, 2004).

Researchers have been exploring innovative approaches to address these challenges and further advance the field of AEE. The substantial progress achieved in face emotion recognition using deep learning is a promising development in this ongoing effort.

The primary objective of this study was to evaluate facial emotion recognition accuracy using sophisticated deep-learning models, including ResNet18, VGGNet16, VGGNet19, and EfficientNet-B0. To compare the performance of these complex models with the baseline convolutional neural network (CNN) model, we used the FER2013 and RAF-DB datasets, focusing on facial expressions and categorizing them into seven primary emotions: anger, disgust, fear, happiness, sadness, surprise, and neutrality. Our findings suggest that more advanced AEE systems can be developed by implementing more complex deep-learning techniques.

The rest of the paper is structured as follows: Chapter 2 presents related research and the literature review. Chapter 3 describes the methodology of the study. Chapter 4 presents and analyzes the results of the study. Finally, conclusions and future directions are provided. This chapter contains general evaluations.

2. RELATED WORK

In recent years, algorithms based on deep neural networks (DNNs) have become more extensively used in image and video analysis. CNNs, including Residual Neural Network (ResNet), VGGNet, and AlexNet have proven to be successful in image classification and feature extraction (Simonyan & Zisserman, 2014; Krizhevsky et al., 2017).

Automatic facial behavior analysis involves tasks such as recognizing basic expressions, estimating continuous emotions, and detecting facial action units, and has advanced significantly due to large-scale datasets and DNNs. For example, Kollias et al., (2019) introduced FaceBehaviorNet, a comprehensive multitask, multidomain, and multilabel network that outperforms single-task networks (Kollias, Sharmanska, 2019; Zafeiriou, 2019). They also proposed strategies for coupling tasks during training, highlighting the advantages of this approach. Furthermore, they have released the Aff-Wild2 dataset, which is the first extensive in-the-wild dataset to provide detailed annotations for each of the three primary behavior tasks (Kollias & Zafeiriou, 2019).

Some researchers exclusively use deep convolutional neural networks to classify basic facial emotions. In contrast, others integrate an additional memory cell aggregating multiple frames to predict current arousal and valence. The FER2013 dataset is used for their research in the following studies. Zhang et al., (2015) used a multi-merged dataset approach for FER2013. They expanded their dataset by including three additional datasets: AFLW, Celeb-faces, and FER2013 datasets. These databases contained labelled facial attributes and the authors introduced a bridging layer that connected the output with the FER2013 dataset to use the features across these datasets. Their method achieved an accuracy of 70.6% in FER2013.

Devries et al., (2014) developed a method for improving FER2013 by accurately assessing the location and shape of facial landmarks. Their models include three CNN layers, a fully connected layer with Relu activation, and an output layer using the L2SVM activation method. Data augmentation techniques such as mirroring, rotating, zooming, and random photo rearrangement were incorporated to enhance their results and this approach attained a FER2013 accuracy of 67.21%.

The RAF-DB dataset, which we also use in this study, has been used in several studies. For example, Li et al., (2017) proposed an approach that introduced a new deep locality-preserving CNN (DLP-CNN) algorithm designed to maximize inter-class scatters while preserving locality closeness to enhance the discriminative ability of deep features, which achieved an accuracy of 74.2% in RAF-DB.

Li, Yong, et al., (2018) introduced an innovative CNN with an attention mechanism (ACNN) designed to detect occluded parts of the human face and emphasize relevant unclouded regions. ACNN uses adaptive weights based on importance and unobstructedness in different facial regions, offering two variants, gACNN (global-local-based ACNN) and pACNN (patch-based ACNN), to consider various regions of interest. The gACNN variant achieved the highest accuracy of 85.07% on the RAF database by integrating global and local representations.

Wang et al., (2020) presented a region attention network (RAN) to adaptively assess the significance of facial areas for emotion recognition, particularly occlusion and position variations. By aggregating area features and employing a region-biased loss, they achieved a notable accuracy of 86.90% on the RAF-DB dataset, showcasing the effectiveness of their RAN model in the FER2013 dataset.

3. METHODS

3.1 Datasets

In our experimental study, we used two distinct datasets (FER2013 and RAF-DB), to train and test models to categorize facial emotions into anger, disgust, fear, happiness, sadness, surprise, and neutrality. The FER2013 dataset consists of grayscale face images, each measuring 48x48 pixels. These images are automatically aligned, ensuring that each face is roughly centered and occupies a consistent portion of the frame. The dataset contains 7,178 examples in the test set and 28,709 examples in the training set. The objective was to categorize each face into one of seven emotion classes based on the emotion it expresses.

The RAF-DB dataset contains 29,672 real-world facial photos with annotations for simple and complex expressions. In our study, we specifically used a subset of 12,271 images labeled with fundamental emotions for the training portion of our research and 3,008 images as test data.

The proposed approach framework is presented in Figure 1. The FER2013 and RAF-DB open-source datasets were used by applying the same pipeline for each data independently. Firstly, the datasets were loaded and split into validation and training sets and the test set was given in each dataset. A robust data loader read the dataset and shuffled it when it was required for training. Different pipelines for augmentation techniques were used to explore the training datasets and decrease the risk of overfitting. Here, we describe the augmentation pipeline and provide practical studies of the application of each hyperparameter we used.

3.2 Deep learning approaches and CNN architectures

To obtain a robust model that could achieve reasonable accuracy, different deep-learning models were employed: CNN as a base model, ResNet18 (He et al., 2016), VGGNet16 and VGGNet19 (Simonyan & Zisserman, 2014), and EfficientNet-B0 (Tan & Le, 2019). Furthermore, we created our own baseline model to serve as a reference for future development and to investigate the effect of transfer learning and the use of widely used deep-learning architectures.

CNNs are sufficient models for various applications (Wayman et al., 2005; Minaee et al., 2019; Yang et al., 2016). VGGNet and Inception models (Szegedy, 2017) showed that raising the network depth increases the quality of model architecture, allowing it to learn faster. By managing the distribution of each layer input, batch normalization (Ioffe et al., 2017) provided learning stability in deep networks and generated more reasonable optimization surfaces. ResNet models (He, Zhang, 2016) showed that identity-based skip connections enabled them to train deeper and more robust networks. The proposed approach framework is presented in Figure 1.

The VGGNet16 model scored 92.7% on the top 5 criteria in ImageNet competition, a database with more than 14,000,000 images separated into 1,000 classes. This was one of the most common models submitted to the ILSVRC-2014 competition. It improves the AlexNet model by sequentially replacing several 1x1 and 5x5, or 3x3 kernel-sized filters with smaller 3x3 kernel-sized filters in the initial and second CNN layers. The NVIDIA Titan Black GPUs were used for the weeks-long training of VGGNet16. Figure 2 presents the architecture of VGGNet16. The input was an RGB image with a defined size of 224x224, which serves as the Cov1 layer's input. The image was run through a stack of CNN layers constructed of a 3x3 kernel size, which is the most suitable size to capture the concepts of each side and center on the input image. The 1x1 CNN layers might be considered the linear transformation of the input channels. The 3x3 CNN layer's spatial padding ensured that the pixel size was preserved after CNN. The padding was 1 pixel, and the stride was also 1 pixel. Furthermore, spatial pooling used max-pooling layers with a 2x2 kernel, and the stride was set to 2.

Two open-source datasets, FER2013 and RAF-DB, were used for this study. We applied the same pipeline for each data set independently. In this study, we applied multiple deep-learning architectures, such as ResNet, EfficientNet-B0, and VGGNet.

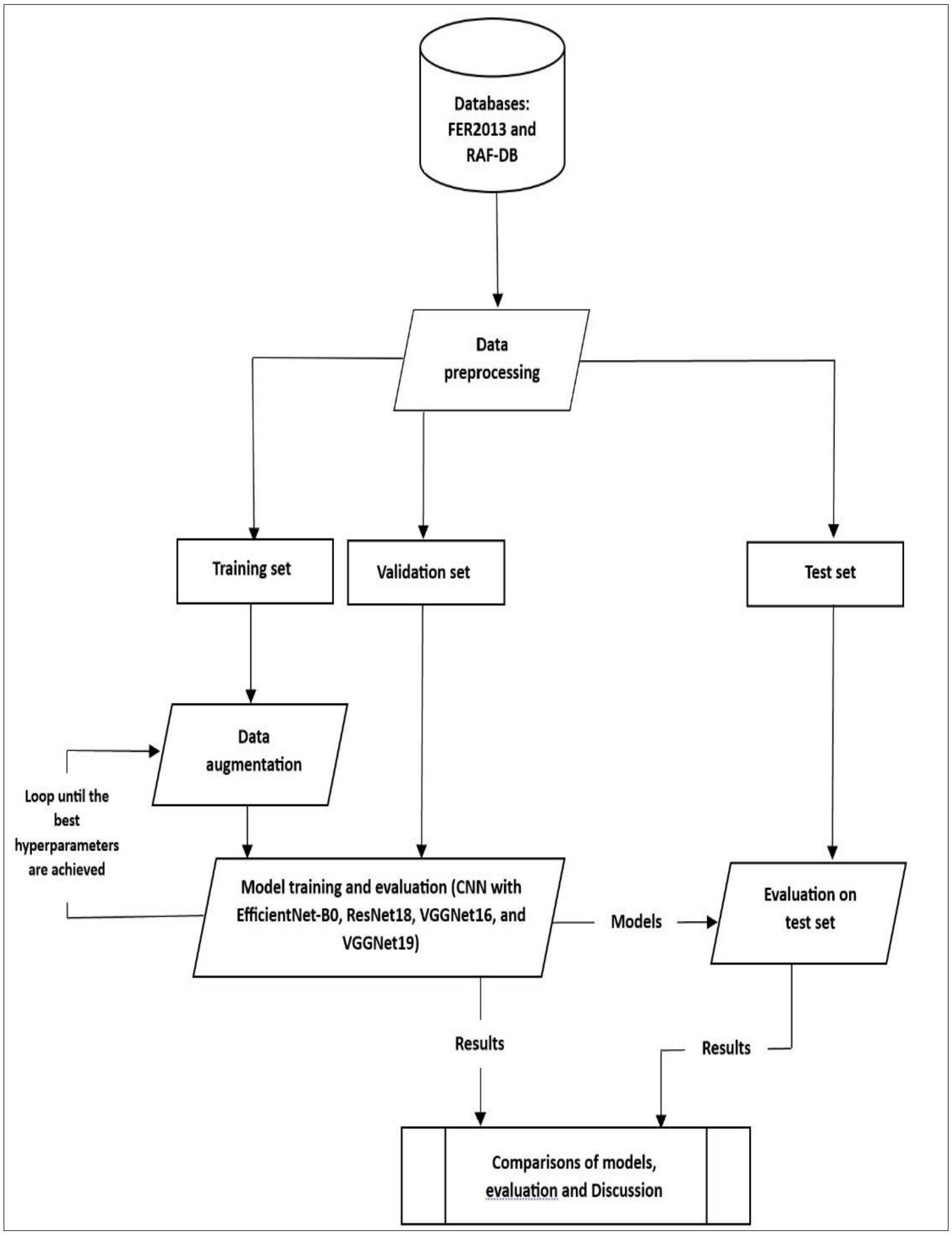

Databases: FER2013 and RAF-DB
Data preprocessing
Training set
Validation set
Test set
Data augmentation
Loop until the best hyperparameters are achieved
Model training and evaluation (CNN with EfficientNet-B0, ResNet18, VGGNet16, and VGGNet19)
Models
Evaluation on test set
Results
Results
Comparisons of models, evaluation and Discussion

Figure 1: Pipeline framework

After the stack of CNN layers, which have varying depths in different designs, there are three fully connected layers; the first two consist of 4,096 neurons, while the third consists of 1,000 neurons for label classification; therefore, has 1,000 neurons representing the number of classes. The softmax layer was the last layer in the model that outputs a probability range from 0 to 1. Figure 3 shows an overview of the VGGNet19 architecture, a CNN used for image classification. VGGNet19 comprises 19 layers, including 16 convolutional layers and three fully connected layers. The convolutional layers are arranged in groups of two or three, with max-pooling layers in between. VGGNet19 has a very deep architecture, allowing it to capture complex image features. The figure highlights the input and output sizes of each layer in the network and the number of filters and kernel sizes used in each convolutional layer. Overall, VGGNet19 is a robust CNN architecture that has been used in a wide range of computer vision tasks.

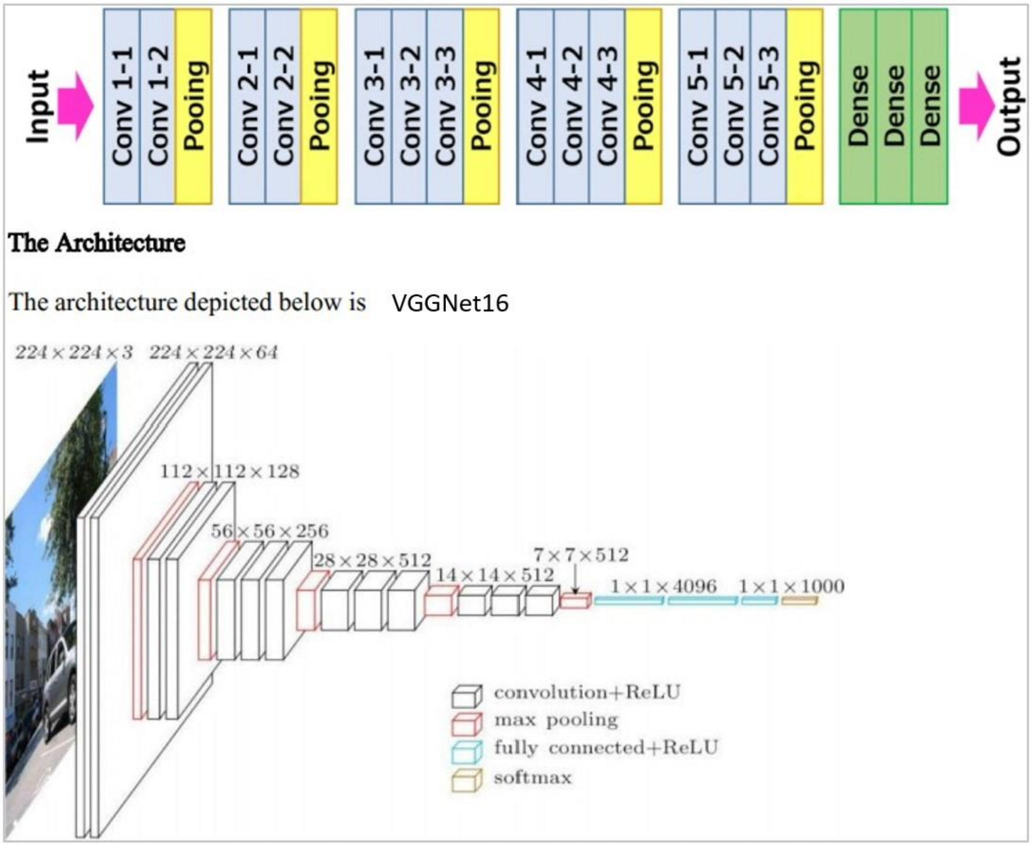


Figure 2: Overview of VGGNet16 architecture

3.3 Data augmentation

Data augmentation methods alter the training data to modify the image pixel representation while keeping the label the same as before. Various methods can be used for augmentation, such as horizontal and vertical flips, color hiccups, random crops, translations, rotations, and many other frequently used augmentations. We were able to double and enlarge our training examples and build a more robust model by applying only a few of these training changes that could notably improve our results. Examples of data augmentation with parameter ranges are listed as follows:

1) Random resize crop: This method is part of the “torchvision.transforms” module. It crops a random area of an image and resizes it to a given size. This method can accept both PIL image and tensor images. The tensor image is a PyTorch tensor with [C, H, W] shape, where C represents the number of channels, and H and W represent the height and width. The method returns a randomly cropped image. Here, we used the scale (0.8 and 1.2), meaning the image was zoomed in by 80% and zoomed out 120%, while keeping the original size (48x48).

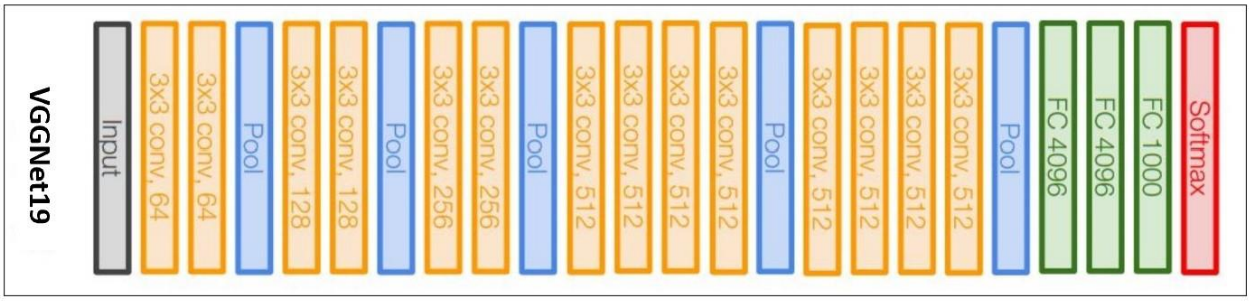


Figure 3: Overview of VGGNet19 architecture

2) Adjust brightness: We randomly changed an image’s brightness, contrast, and saturation. We used brightness=0.5, contrast=0.5, and saturation=0.5, with 50% probability (p=0.5).
3) Random affine: the approach accommodates both PIL and tensor images. The tensor image is a PyTorch tensor with [C, H, W] shape, where C represents the number of channels and H and W represents the height and width. This method returns the affine transformed image of the input image. We used 0, translate=(0.2, 0.2), with 50% probability (p=0.5).

4) Horizontal flip: Also known as image mirroring, this is a transformation that swaps the left and right sides of the image, creating a mirror image effect with a 50% probability.

5) Random rotation ranges from [-45,45] degrees.

6) Tencrop: After all the aforementioned transforms are applied, this method takes the image and returns a tuple of 10 cropped PIL or tensor images. We used (40,40) crop dimensions.

7) Normalization: This is a good practice when using DNNs. Normalizing the images means changing the images to have a mean (0.0) and standard deviation (1.0) for each channel. This helps ensure that the input data has a similar scale and distribution, which can improve the model’s performance. The “torchvision.transforms.Normalize()” method can be used to normalize the images in PyTorch. This method takes each channel’s mean and standard deviation values as inputs and returns a normalized tensor image.

8) Random erasing: A “torch.Tensor” image has its pixels erased in a rectangular region that is chosen randomly. We used a 50% probability.

For example, we applied small ranges of these parameters to be compatible with real-life images in training. It is important to note that this advanced data augmentation pipeline provided the model with a comprehensive learning experience, enabling it to grasp nuanced features and intricacies within the data. We enhanced the model’s adaptability and ability to make accurate predictions across various real-world scenarios by continually refining and optimizing the data augmentation strategies. The augmentation results are illustrated in Figures 4 and 5, showcasing the effects of our augmentation pipeline on the FER2013 and RAF-DB datasets. The left images depict the original samples before augmentation, while the corresponding right images show the same examples after applying our augmentation pipeline.

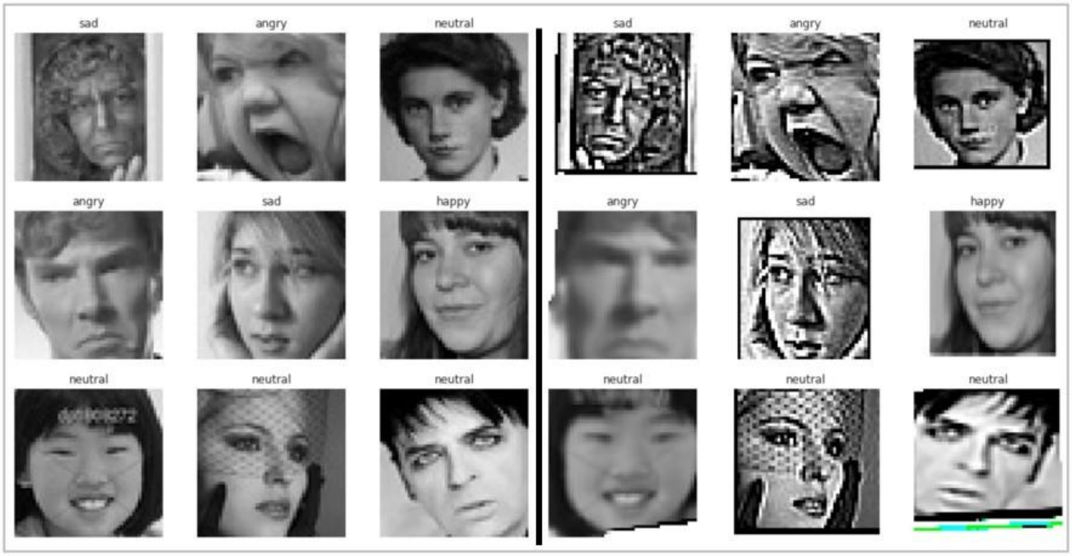


Figure 4: FER2013 dataset before and after applying our augmentation pipeline

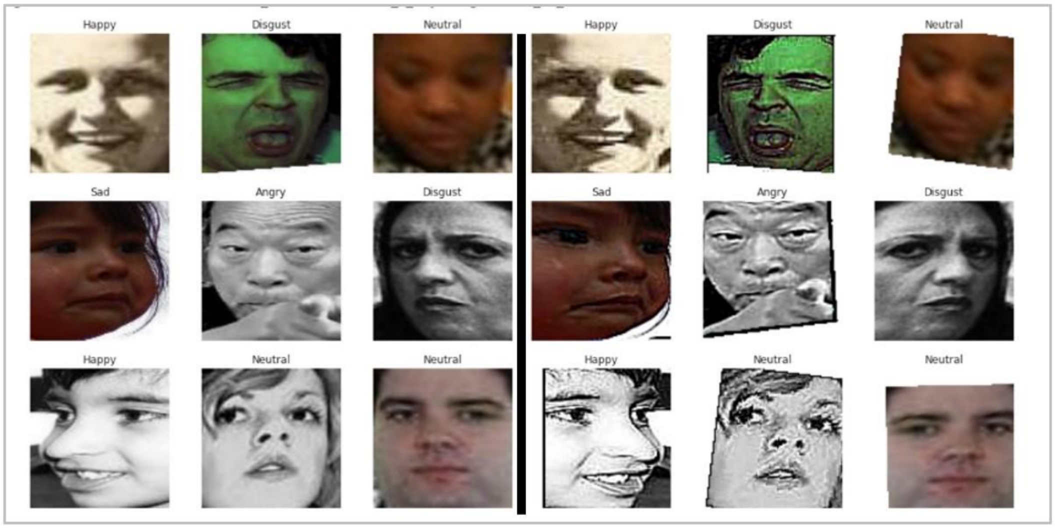


Figure 5: RAF-DB dataset before and after applying our augmentation pipeline

4. RESULTS AND DISCUSSION

4.1 Baseline model

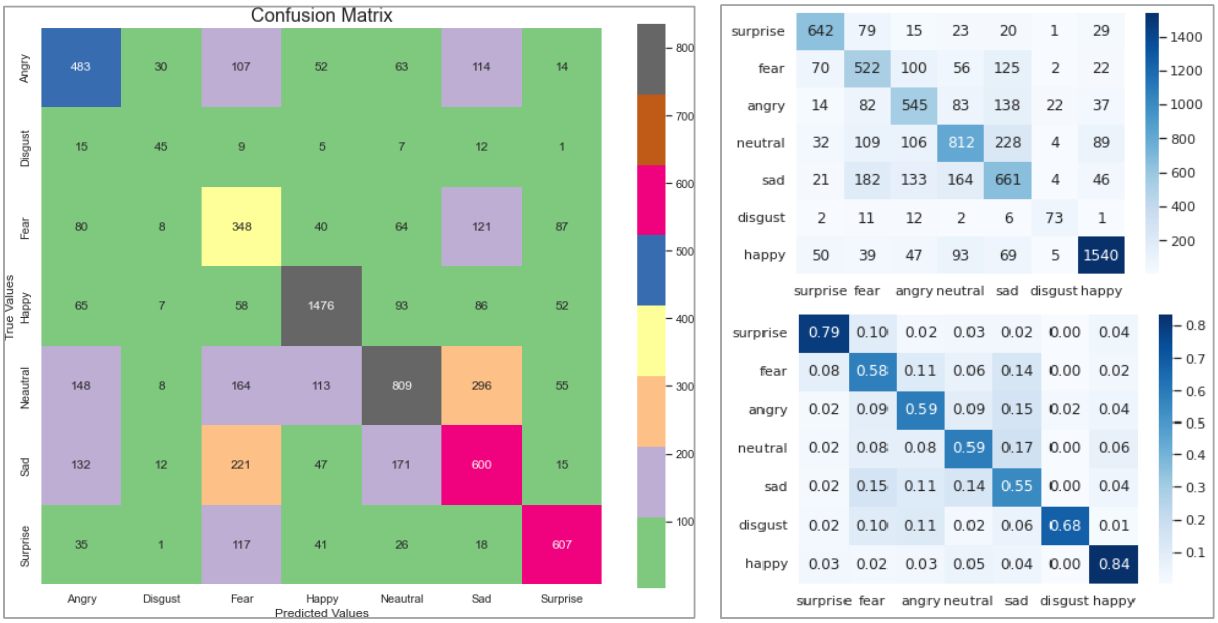
We developed our baseline model to serve as a reference for future development and to investigate the effect of transfer learning on subsequent deep-learning models. The confusion matrix of the baseline model is presented in Figure 6.a. We calculated the precision, recall, and F1-score of the baseline model for each label independently in Table 1. This provides insight into how the model performed on unseen data and where the labels had the most errors. It also allows us to gauge the complexity of the problem at hand. It highlights the significance of leveraging transfer learning and employing more intricate models on the test dataset to enhance its ability to memorize it more effectively.

Table 1: Classification report of the baseline model

Label	Precision	Recall	F1-score
Angry	0.56	0.50	0.53
Disgust	0.48	0.41	0.44
Fear	0.47	0.34	0.39
Happy	0.80	0.83	0.82
Neutral	0.51	0.66	0.57
Sad	0.50	0.48	0.49
Surprise	0.72	0.73	0.72

4.2 EfficientNet-B0

To better understand the performance of the efficient model on the FER2013 and RAF-DB datasets, various metrics were analyzed, including the confusion matrix, normalized confusion matrix, precision, recall, and F1-score for each label. The normalized confusion matrix is the confusion matrix divided by the number of images per label to calculate the error distribution in percent as in Figure 6.b, which shows the confusion matrix and normalized confusion matrix of the efficient model on the FER2013 dataset. The happy (84%) and surprise (79%) labels showed the best accuracy. The model did get confused about the four labels: fear, angry, neutral, and sad, which had the highest error percentage. Table 2 shows the classification report of the model. Furthermore, the number of images that contributed to that result, which provides more intuition about how our efficient model performs on FER2013, is presented. The surprise label received a 0.78 F1-score even if it did not have more images as with neutral and sad images. Figure 7 shows the confusion matrix and normalized confusion matrix of the EfficientNet-B0 model of the RAF-DB test set. RAF-DB had a more robust error distribution because the data has an RGB image with a size of 100x100, enabling the model to capture more information about the data. Table 3 shows the classification report of RAF-DB, which achieved 84.547% accuracy on the test set. The model predicted happy images accurately with a 0.93 F1-score, which was due to the data having many happy examples that enabled the model to predict this label more accurately. Also, the three labels, surprise, sad, and neutral, are above the 0.8 F1 metric score.



(a) (b)

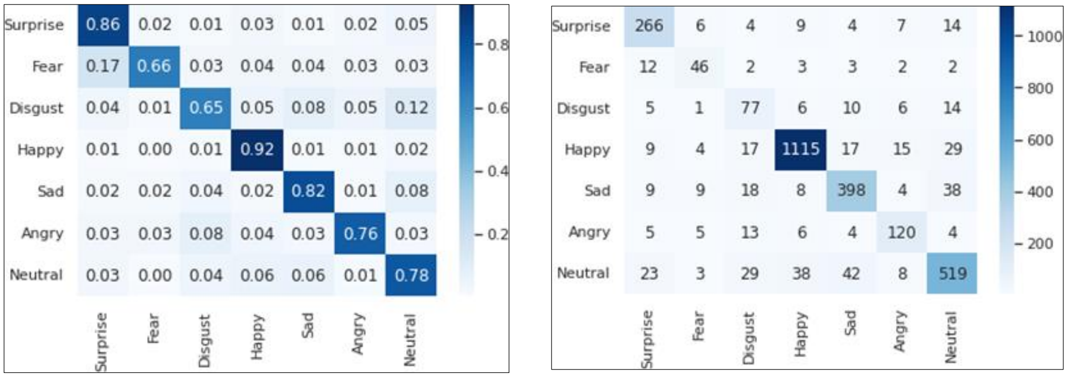
Figure 6: Results for the (a) confusion matrix baseline model and (b) confusion matrix and normalized confusion matrix EfficientNet-B0 on FER2013

Table 2. Classification report of the EfficientNet-B0 model on FER2013

Label	Precision	Recall	F1-score	Images (n)
Surprise	0.79	0.77	0.78	831
Fear	0.58	0.51	0.54	1024
Angry	0.59	0.57	0.58	958
Neutral	0.59	0.66	0.62	1233
Sad	0.55	0.53	0.54	1247
Disgust	0.68	0.66	0.67	111
Happy	0.84	0.87	0.85	1764
Accuracy	-	-	67.09%	7178

Table 3: Classification report of the EfficientNet-B0 model on RAF-DB

Label	Precision	Recall	F1-score	Images (n)
Surprise	0.86	0.81	0.83	329
Fear	0.66	0.62	0.64	74
Disgust	0.65	0.48	0.55	160
Happy	0.92	0.94	0.93	1185
Sad	0.82	0.83	0.83	478
Angry	0.76	0.74	0.75	162
Neutral	0.78	0.84	0.81	620
Accuracy	-	-	84.547%	3008



	Surprise	Fear	Disgust	Happy	Sad	Angry	Neutral
Surprise	0.86	0.02	0.01	0.03	0.01	0.02	0.05
Fear	0.17	0.66	0.03	0.04	0.04	0.03	0.03
Disgust	0.04	0.01	0.65	0.05	0.08	0.05	0.12
Happy	0.01	0.00	0.01	0.92	0.01	0.01	0.02
Sad	0.02	0.02	0.04	0.02	0.82	0.01	0.08
Angry	0.03	0.03	0.08	0.04	0.03	0.76	0.03
Neutral	0.03	0.00	0.04	0.06	0.06	0.01	0.78

	Surprise	Fear	Disgust	Happy	Sad	Angry	Neutral
Surprise	266	6	4	9	4	7	14
Fear	12	46	2	3	3	2	2
Disgust	5	1	77	6	10	6	14
Happy	9	4	17	1115	17	15	29
Sad	9	9	18	8	398	4	38
Angry	5	5	13	6	4	120	4
Neutral	23	3	29	38	42	8	519

Figure 7: Confusion matrix and normalized confusion matrix EfficientNet-B0 on RAF-DB

4.3 ResNet18

The model was evaluated to gain a better understanding of how the model performs on the test set. For each dataset, the confusion matrix and normalized confusion matrix were computed. A classification report was also generated that shows the precision, recall, and F1-score for each of the seven labels. The model predicted the happy and surprise labels with 86% and 80% accuracy, respectively. The ResNet18 model also reduced the error on the disgusting label and achieved 78% accuracy. The error was high on sad and fear labels; also, the model was confused between both labels. Table 4 gives the classification report of the model on the FER2013 dataset. The model achieved a total accuracy of 68.07% on the test set; we observed that the F1-score per each label increased compared with the EfficientNet model. Figure 8 shows the confusion matrix and normalized confusion matrix of the ResNet18 model on the RAF-DB test set. We observed that the model was more robust on this dataset and could reach a reasonable accuracy compared to studies in the literature.

Table 4: Classification report of the ResNet18 model on FER2013

Label	Precision	Recall	F1-score	Images (n)
Surprise	0.80	0.80	0.80	831
Fear	0.54	0.47	0.50	1024
Angry	0.62	0.61	0.61	958
Neutral	0.61	0.66	0.63	1233
Sad	0.55	0.56	0.55	1247
Disgust	0.78	0.66	0.71	111
Happy	0.86	0.87	0.87	1764
Accuracy	-	-	68.07%	7178

Table 5 details how our model performed on the RAF-DB test set; it shows the precision, recall, and F1-score per each label of the seven basic emotions. The model achieved a test accuracy of 86.02% and a 0.94 F1-score for the happy label, and also gets above 0.80 on four labels: surprise, sad, angry, and neutral. Figure 8 shows the confusion matrix and normalized confusion matrix of the ResNet18 model in the RAF-DB dataset.

Table 5: Classification report of the ResNet18 model on RAF-DB

Label	Precision	Recall	F1-score	Images (n)
Surprise	0.86	0.84	0.85	329
Fear	0.67	0.61	0.64	74
Disgust	0.71	0.59	0.65	160
Happy	0.93	0.94	0.94	1185
Sad	0.84	0.85	0.84	478
Angry	0.82	0.78	0.80	162
Neutral	0.80	0.85	0.82	620
Accuracy	-	-	86.02%	3008

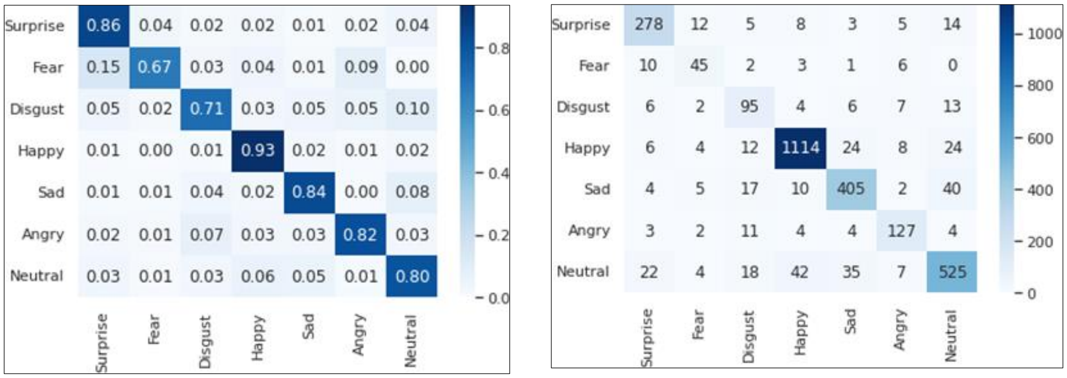


Figure 8: Confusion matrix and normalized confusion matrix ResNet18 on the RAF-DB

4.4 VGGNet16

The VGGNet16 model was evaluated on the test set, but with added customization layers to this model. Firstly, we changed the last average pooling layer output size from (7,7) to (1,1), meaning we used global average pooling instead. Furthermore, changing the top layer to have seven output neurons to represent our seven classes determined the confusion matrix and normalized confusion matrix. The VGGNet16 model achieved appropriate results with the FER2013 dataset, with a 70.2% test accuracy. VGGNet's original model has 138,000,000 parameters, and this number was too big compared with other models, and the complexity was too high, but the customization we added reduced the number of parameters to 33,600,000 parameters, which was more viable. Figure 9 shows the model's confusion matrix and normalized confusion matrix on the FER2013 test set. The VGG16 model reduced the error on all labels compared with other models based on the diagonal of the VGGNet confusion matrix. Table 6 shows the precision, recall, and F1 scores for all labels and the number of images per label that contributed to the calculated metrics. The F1-score of each label increased and the overall accuracy on the test set was 70.2%. We also show the confusion matrix and normalized confusion matrix of the VGGNet16 model on the RAF-DB in Figure 10. The ResNet18 model achieves the best result on this dataset, but since the VGGNet16 model achieves 85.88% while ResNet18 gets 86.02%, the difference in accuracy was not great; however, it is important to note that the VGGNet16 model had more parameters, and its complexity was too high. Table 7 shows more details about how the VGGNet16 model performed on each label.

Table 6: Classification report of the VGGNet16 model on FER2013

Label	**Precision**	**Recall**	**F1-score**	**Images (n)**
surprise	0.82	0.81	0.82	831
fear	0.56	0.52	0.54	1024
angry	0.64	0.62	0.63	958
neutral	0.64	0.70	0.67	1233
sad	0.57	0.57	0.57	1247
disgust	0.85	0.68	0.75	111
happy	0.88	0.90	0.89	1764
Accuracy	-	-	70.2%	7178

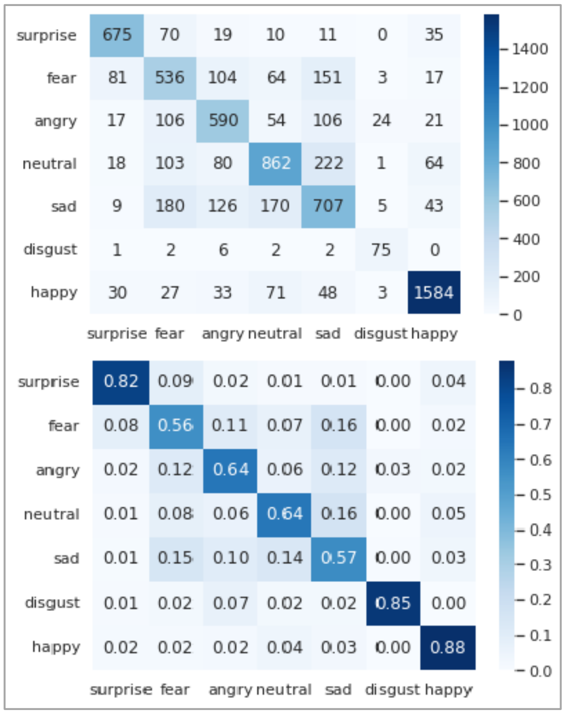


Figure 9: Confusion matrix and normalized confusion matrix VGGNet16 on FER2013

Table 7: Classification report of the VGGNet16 model on RAF-DB

Label	Precision	Recall	F1-score	Images (n)
Surprise	0.86	0.85	0.85	329
Fear	0.64	0.57	0.60	74
Disgust	0.59	0.57	0.58	160
Happy	0.94	0.95	0.95	1185
Sad	0.83	0.82	0.83	478
Angry	0.80	0.75	0.78	162
Neutral	0.81	0.84	0.83	620
Accuracy	-	-	85.88%	3008

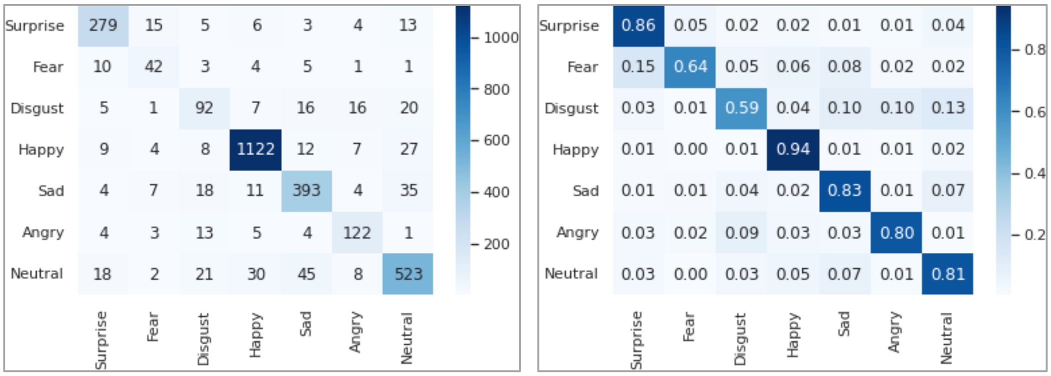


Figure 10: Confusion matrix and normalized confusion matrix of the VGGNet16 model on RAF-DB

4.5 VGGNet19

The VGGNet19 model was evaluated on the test set. Firstly, we changed the last average pooling layer output size from (7,7) to (1, 1), meaning that we used global average pooling instead. Also, the top layer was changed to have seven output neurons to represent our seven classes. This customization led to a reduction in the number of parameters that exceed 138,000,000 parameters to just 45,200,000 parameters.

The VGGNet19 model achieved the best result for the FER2013 dataset; it achieved a 71.02% test accuracy. The model reduced the error on all labels compared with the other models, which we observed by comparing the diagonal of the VGGNet19 confusion matrix with the other models. We found that the F1-score of each label increased, and that the overall accuracy on the test set was 71.02% (Table 8). Table 9 provides more information on how the VGGNet19 model performed on each label.

Table 8: Classification report of the VGGNet19 model FER2013

Label	**Precision**	**Recall**	**F1-score**	**Images (n)**
surprise	0.84	0.79	0.82	831
fear	0.57	0.59	0.58	1024
angry	0.61	0.64	0.63	958
neutral	0.68	0.67	0.67	1233
sad	0.59	0.57	0.58	1247
disgust	0.69	0.72	0.70	111
happy	0.89	0.90	0.89	1764
Accuracy	-	-	71.02%	7178

Table 9: Classification report of the VGGNet19 model on RAF-DB

Label	**Precision**	**Recall**	**F1-score**	**Images (n)**
Surprise	0.87	0.83	0.85	329
Fear	0.77	0.50	0.61	74
Disgust	0.66	0.62	0.64	160
Happy	0.92	0.95	0.94	1185
Sad	0.84	0.83	0.83	478
Angry	0.87	0.77	0.82	162
Neutral	0.81	0.85	0.83	620
Accuracy	-	-	85.87%	3008

4.6 Model comparisons

Here we compare the results of all models based on test accuracy. Four different deep CNN architectures were used: ResNet18, EfficientNet-B0, VGGNet16, and VGGNet19 to test the FER2013 dataset. The accuracies of these deep learning approaches were promising and was further validated on the RAF-DB dataset. Table 10 shows the overall results of all models on the FER2013 dataset. The deep learning models had the best accuracy compared with the baseline model. VGGNet19 achieved the highest accuracy (71.02%) for FER2013. However, the number of model parameters and the model's complexity should be considered. Table 11 presents the overall results on RAF-DB; based on the FER2013 results, we find that the three deep learning models are promising for solving the problem of emotion recognition; therefore, we only used these deep learning models on the RAF-DB. The ResNet18 model achieved the best accuracy (86.02%) for this dataset.

Table 10: All model test results for FER2013

Model	**Test Accuracy**	**Num. of Parameters**
Base Line	60.854%	1,672,775
EfficientNet-B0	67.09%	4,016,515
ResNet18	68.07%	11,180,103
VGGNet16	70.20%	33,625,927
VGGNet19	71.02%	45,227,079

Table 11: All model test results for RAF-DB

Model	**Test Accuracy**	**Num. of Parameters**
EfficientNet-B0	84.55%	4,016,515
ResNet18	86.02%	11,180,103
VGGNet16	85.88%	33,625,927
VGGNet19	85.87%	45,227,079

4.7 Study robustness

To confirm the robustness of our approach, we applied our pipeline to two datasets. Furthermore, we avoided depending solely on a single algorithm to solve the problem and instead explored different approaches and models, aiming to develop resilient models that accurately represents our study. To acquire the model's hyperparameters and training augmentation pipeline, we performed an exhaustive search between various pipelines to find robust hyperparameters that gave us the best result representative of each model independently. We ensured the robustness of our approach by applying our pipeline to two open source datasets. By comparing our results with published data, our model displayed better performance and achieve better results on the datasets. Tables 12 and 13 compare our best results with different state-of-the-art model that have been previously applied to both datasets. The accuracy of VGGNet19 on FER2013 and ResNet18 on RAF-DB outperform all other models. Some researchers depend on multiple datasets to merge them and construct one bigger dataset; however, in our study we relied only on each dataset independently. The customization layers we used for each model decreased the number of parameters of the models, and the inference time also increased compared with the published models.

Table 12: Comparison of models used to test on FER2013

Models	Accuracy for test set
Devries et al.	67.21%
Zhang et al.	70.60%
Ours (VGGNet19)	**71.02%**

Table 13: Comparison of models used to test on RAF-DB

Models	Accuracy for test set
gACNN	85.07%
DLP-CNN	74.20%
Ours (ResNet18)	**86.02%**

6. CONCLUSION

In this study, we have independently assessed each model's strengths and weaknesses and performed a comprehensive comparison to offer concise discussions and distinctions among them. Moreover, we have shown that CNN architectures such as ResNet, EfficientNet, and VGGNet models can achieve good results for emotion recognition using different adjustments. These considerations should be taken into account for future enhancements in this study. The models were evaluated on the test sets to gain a better understanding of how they work when dealing with unobserved data. For each dataset, the confusion matrix and normalized confusion matrix were computed. On the FER2013 dataset, the VGGNet19 model produced the best results, scoring a test accuracy of 70.2%. The 138,000,000 parameters in the original VGGNet model make it overly complex and large compared with other models; however, the customization we added reduced the number of parameters to 33,600,000 M, which was more efficient. The CNN was used on the FER2013 test set and studies of the residual learning blocks have been shown to increase the performance of deep learning models. We selected the number of layers and neurons based on our intuition of applying CNNs for different datasets. We used trial-and-error to determine the best learning rate and scheduler. Our proposed model can be combined with other models to increase the accuracy of face recognition systems that use the FER2013 dataset because it is computationally less expensive. Although the FER2013 dataset is very complicated, with a small number of samples in some classes, the number of samples in each class can be increased by the correct amount to improve accuracy. To solve the problem, we used transfer learning techniques to retrain VGGNet16, VGGNet19, ResNet18, and EfficientNet-B0 on two open sources of datasets, FER2013 and RAF-DB. Using pre-trained deep CNN architectures showed great performance with these datasets. The VGGNet19 model performed the best on FER2013 with added customization layers. We changed the last average pooling layer output size from (7,7) to (1,1), to use global average pooling. Furthermore, we changed the top layer to have seven output neurons representing the seven classes. We also significantly reduced the number of VGGNet's original parameters by approximately 60%, which substantially improved the performance. ResNet18 outperformed all other models with 86.02% accuracy on the RAF-DB dataset. In this study, different CNN models applied in studies on the problem of emotion classification from facial expressions were comprehensively analyzed and the effectiveness of the models for realizing effective approaches in this field was presented with their advantages.

REFERENCES

Bartlett, Marian Stewart, et al. "Machine learning methods for fully automatic recognition of facial expressions and facial actions." 2004 IEEE International Conference on Systems, Man and Cybernetics (IEEE Cat. No. 04CH37583). Vol. 1. IEEE, 2004.

Hudlicka, Eva. "To feel or not to feel: The role of affect in human–computer interaction." International journal of human-computer studies 59.1-2 (2003): 1-32.

B Allen & Pease. The definitive book of body language, 2004.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556, 2014.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. Communications of the ACM, 60(6):84–90, 2017.

Dimitrios Kollias, Viktoriia Sharmanska, and Stefanos Zafeiriou. Face behavior a la carte: Expressions, affect and action units in a single network. arXiv preprint arXiv:1910.11111, 2019.

Dimitrios Kollias and Stefanos Zafeiriou. Expression, affect, action unit recognition: Aff-wild2, multi-task learn- ing and arcface. arXiv preprint arXiv:1910.04855, 2019.

Zhang, Zhanpeng, et al. "Learning social relation traits from face images." Proceedings of the IEEE International Conference on Computer Vision. 2015.

Terrance Devries, Kumar Biswaranjan, and Graham W Taylor. Multi-task learning of facial landmarks and expression. In 2014 Canadian conference on computer and robot vision, pages 98–103. IEEE, 2014.

Li, Shan, Weihong Deng, and JunPing Du. "Reliable crowdsourcing and deep locality-preserving learning for expression recognition in the wild." Proceedings of the IEEE conference on computer vision and pattern recognition. 2017.

Li, Yong, et al. "Occlusion aware facial expression recognition using CNN with attention mechanism." IEEE Transactions on Image Processing 28.5 (2018): 2439-2450.

Kai Wang, Xiaojiang Peng, Jianfei Yang, Debin Meng, and Yu Qiao. Region attention networks for pose and occlusion robust facial expression recognition. IEEE Transactions on Image Processing, 29:4057–4069, 2020.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In Proceedings of the IEEE conference on comp. vision and pattern recognition, p. 770–778, 2016.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556, 2014.

Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In International conference on machine learning, pages 6105– 6114. PMLR, 2019.

James Wayman, Anil Jain, Davide Maltoni, and Dario Maio. An introduction to biometric authentication systems. In Biometric Systems, pages 1–20. Springer, 2005.

Shervin Minaee, Amirali Abdolrashidi, Hang Su, Mohammed Bennamoun, and David Zhang. Biometrics recognition using deep learning: A survey. arXiv preprint arXiv:1912.00271, 2019.

Shuo Yang, Ping Luo, Chen-Change Loy, and Xiaoou Tang. Wider face: A face detection benchmark. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 5525–5533, 2016.

Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In Thirty- first AAAI conference on artificial intelligence, 2017.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In International conf. on machine learning, p. 448–456. PMLR, 2015.

AUTHORS

Ahmed Adnan Hameed Qutub: Obtained his degree in Computer Sciences from the University of Kirkuk, Iraq, in 2007. He is currently studying for a Master's degree and is a researcher at Gazi University Ankara, Türkiye. His lines of research are related to information technology, computer vision (specifically with deep learning), and using artificial intelligence in medical diagnosis with Python.

Yılmaz Atay received his PhD from the Department of Computer Engineering, Selcuk University, Türkiye, in 2018. He was a Bioinformatics laboratory researcher at the Department of Computer and Information Science and Engineering, University of Florida, USA, in 2016 and is currently working as a faculty member at the Department of Computer and Engineering at Gazi University. His research focuses on complex network analysis, graph-based approaches, bioinformatics, and machine and deep learning.