RECOGNIZING FACIAL EMOTION IN REAL-TIME USING MUWNET A NOVEL DEEP LEARNING NETWORK

Mustafa Mohammed Kataa Al-Khazaali, Wandeep Kaur

Fakulti Teknologi dan Sains Maklumat, Universiti Kebangsaan Malaysia
43600 UKM Bangi, Selangor Malaysia.

mustafarh0@gmail.com, wandeep@ukm.edu.my

ABSTRACT

Facial expression recognition (FER) is a branch of psychology that studies the classification of human emotions using facial expressions. Particularly, FER can be implemented in a vast array of applications, including education, online entertainment, and even essential fields involving human lives and behavior, such as medicine. There are seven universal facial expression categories: surprise, sadness, happiness, contempt, fear, anger, and neutrality. Recognizing all these facial expressions and predicting a person's present mood is a challenging problem for machines. Because of the nature of humans, this challenge presents itself to a computer in a more sophisticated manner. The main objective of the thesis was to construct a novel deep Convolutional Neural Network (CNN) for facial expression classification that can assist in extracting features from images to identify facial gestures and then apply it in real-time. Various neural network models and classification methods have been introduced in the past to reach cutting-edge accuracy in this industry. Separate studies have investigated the capabilities and effectiveness of CNN models in distinguishing human emotions on the FER2013 dataset. In this study, the proposed MuWNet model has been diversified with several types of layers, such as convolution layers, separable convolution layers, and residual blocks. In addition, applying hyperparameter tweaking to enhance progress. The results of two experiments that have been done on the MuWNet model indicate that the accuracy of the classification in the second experiment was 70.72%, with an increase of 0.14% over the first. Finally, these results appear to be competitive in the field of FER, and it can be stated that the proposed model contributed to the emergence of a classification system for facial expressions.

Keyword: Facial Emotion recognition, Convolutional neural network, Deep learning, FER2013 Dataset

I. INTRODUCTION

Emotions are a person's means of expressing their sentiments since people can communicate with one another either verbally or non-verbally. Hence, due to advances in technology, computer hardware, and graphics processing units (GPUs), the demand for human-computer interaction (HCI) has grown in recent decades. Researchers are now able to establish or construct a powerful artificial intelligence (AI) system that can automate a person's actions in a variety of industries.

The face expresses an individual's identity. Categorizing emotions through facial expressions can be more accurate than speaking or gesturing. Joy, neutrality, anger, fear, sadness, disgust, and surprise are seven facial expressions used by the authors (Pathar et al. 2019) to classify people's emotions. Similarly, in some cases, the interaction of the mouth, cheeks, eyes, brows, and front face could reveal more information about human feelings than words (Kaviya & Arumugaprakash 2020). Herein lies the importance of using facial emotion recognition (FER) in commerce, health, and education. Even though building a model to detect emotions is challenging, (Khaireddin & Chen 2021) stated that applying convolutional neural networks (CNNs) to this task could surpass other models that use classical image processing methods owing to the ability of CNNs to extract features from images and the effectiveness of their computation. Furthermore, when compared to traditional machine learning (ML) models such as support vector machines (SVM), CNN can produce high accuracy results (Gaddam et al. 2022).

In this research, a face recognizer will be implemented using deep learning neural networks. Finally, a real-time face recognition system will be used to identify human expressions in two phases. The first phase is face localization, which involves finding a face in an image or video. The second stage involves categorizing facial expressions into one of seven groups.

Building a human face expression recognizer using deep learning (DL) neural networks is a difficult task, as there are many important factors to consider, such as storage size, number of parameters, and layer level in a DL model, all of which can affect performance in real-time applications. When using several layers, as in AlexNet and VGGNet, which have a very deep structure, the complexity and size of CNNs grow, posing issues for real-time systems (Cotter 2020). This study tries to address the issue of facial emotion identification by developing a novel CNN model called MuWNet. While attempting to get a comparable accuracy outcome to state-of-the-art models.

II. LITERATURE REVIEW

This current study focuses on facial emotion recognition (FER), which is the process of determining which expression is employed in a captured image or video. The review emphasizes the importance of comprehending human emotions. The vast majority of FER approaches incorporate the utilization of CNN and machine learning. Many different algorithms have also been used for a broad variety of datasets, such as FER-2013, the CK+, the RaFD, the JAFFE datasets, and many others.

Sang et al. (2017) developed multiple methods for identifying facial expressions in humans. The methods are dependent on CNN. Their techniques are influenced by the VGG design principles. In facial expression recognition, the authors found that L2 multi-class SVM loss is preferable to cross-entropy loss by comparing them on the FER2013 dataset using different loss functions. BKVGG12, which consisted of 12 layers, was the model with the highest level of accuracy, coming in at 71.9%.

Using the FER-2013 dataset, Agrawal and Mittal (2019) built two CNN algorithms. In addition, they evaluated the influence of CNN parameters, notably kernel size and filters' number, on the classification precision. Their work made a substantial contribution by testing a variety of kernel sizes along with filters to propose two new CNN architectures with a 65% human-like accuracy. According to their research, the kernel size and filters' number were found to have a substantial effect on the network's accuracy. In addition, the accuracy of both proposed models exceeded 65%.

Bhandari and Pal (2021) investigated whether the use of edges can help CNN identify emotions from images. To identify facial expressions from photographs, a CNN model consisting of two towers and accepting a variety of inputs has been developed. Accordingly, they reasoned that edges in an image provide discriminatory information and that their explicit usage is predicted to assist in the training of CNNs and better emotion recognition. This is because their explicit use is expected to help improve accuracy. Their proposed CNN included two inputs. The first input was the image itself, while the second was the edge image acquired by the Canny edge detector. The researchers tested their findings on two datasets, JAFEE and FER2013, to show that using edge information explicitly enhances classifier performance. The proposed model attained an accuracy of 85.7% on the JAFEE dataset and 63.7% on the FER2013 dataset.

Vulpe-Grigorasi and Girgore (2021) aimed to increase the efficiency of a CNN network by fine-tuning its architecture and hyperparameters to classify human facial images into distinct emotional categories. They did this by using images of people's faces. The ideal hyperparameters were identified by creating and training models utilizing a random search technique applied to a search space containing discrete hyperparameter values. To prove that an effective solution may be discovered in a search space where previous results are considered to be local minima, the researchers wanted to optimize the hyperparameters of the model superficially. In addition, the only layers of the model architecture that were optimized were the convolutive layers; the categorization layers were left out of the optimization process. The most accurate model was able to attain a score of 72.16 % after being trained and evaluated using the FER2013 database.

As several applications respond to the emotional state of a participant, it is crucial to establish an accurate FER on a smartphone. Cotter (2020) introduced MobiExpressNet, a novel light deep-learning model for FER based on two frameworks, MobileNetV1 and MobileNetV2. The researcher began by utilizing a series of kernels and then moved on to utilizing a series of depth-wise convolution filters, to extract the feature maps. The investigation led to the discovery by the researcher that the best network model has an accuracy of 67.96% of the challenging FER2013 dataset. This is 2.5% more accurate than human accuracy. In addition, it was discovered that the

MobiExpressNet model was more than five times smaller than the smallest MobileNet model, which makes the new model particularly interesting for use in real-time applications.

III. RESEARCH METHODOLOGY

The methodology utilised in this study is broken down into four distinct phases, the first of which entails the preparation of the FER2013 dataset in addition to its preliminary processing. The second phase is known as the training phase, and it is during this phase that the suggested model is trained and modifications to the ideal parameters are determined. In the third phase, the model that was created will be evaluated using the data that is currently available. To assess its performance and acquire the level of precision required, it is necessary to compare the proposed model to the research that has come before it. The final phase of this research will involve putting the model into action in real-time to determine whether or not it can assist in the identification of human emotions. Figure 1 provides a graphic representation of the study design.

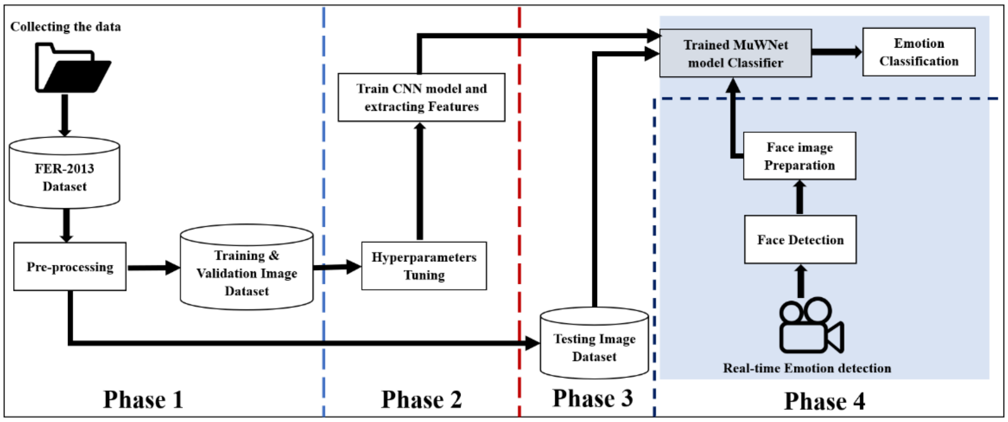

Figure 1 Graphical representation of the proposed system

As previously indicated, this research is separated into four phases. The first phase consists of two steps. The first involves collecting and preparing the data, while the second entails pre-processing the data and separating it into training, validation, and testing datasets.

The second phase includes three steps the initial step entails collecting the training data. The subsequent step is hyperparameter tuning, which involves locating the optimal hyperparameters to enhance model accuracy on image data. In the final step, CNN, one of the DL methods, will be utilized to train the model using an image training dataset. CNN will assist in the extraction of characteristics to be included in the proposed model, removing the need to manually extract them.

The third phase employs the testing data images to gauge the model's performance and compare its accuracy to that of similar works. This phase is comprised of two distinct parts. Having available test data to feed into the proposed model is the initial step, followed by assessing the mentored model on test images and determining its prediction accuracy.

In the fourth phase, the MuWNet model will be applied in real-time to assist in recognizing human emotions. This might be accomplished by first employing the Haar Cascade classifier, a well-known face recognition technique, which will be utilized in this study to capture a human face from a video frame and crop the face so that it can be fed on the suggested model, Secondly, a face image preparation will be conducted to aid in resizing the cropped face image so that it can be given as input to the proposed model. Finally, the MuWNet model that was trained and tested in phases 2 and 3 is now ready to be provided with the resized image, and the outcome is then the category to which the emotion belongs.

1- DATASET

The FER2013 dataset used in this study was created by Carrier and Courville (2013) as part of a larger research project and was used in one of Kaggle's representation learning competitions. The FER2013 data set includes grayscale facial images with 48 by 48 pixels and seven numerically ordered classes, where Angry is represented by 0, Disgust by 1, Fear by 2, Happy by 3, Sad by 4, Surprise by 5, and Neutral by 6. The training set comprises 28,709 distinct examples for each of the seven categories. For the leaderboard, a public test set consisting of 3,589 samples was employed. Furthermore, to identify the contest's winner, an extra 3,589 samples were incorporated into the final private test set.

2- MUWNET MODEL ARCHITECTURE

This experiment aims to develop the MuWnet model so that it can recognize seven distinct facial emotions.

The deep neural network that is a component of the MuWNet model drew its motivation from that of the VGGNet network (Simonyan & Zisserman 2014), the ResNet network (He et al. 2016), and the MobileNet (Howard et al. 2017). Also, the MuWNet model was named after the student and supervisor for this study, with Mu being the first two letters of Mustafa and W being the first letter of Wandeep. The structure of the MuWNet CNN model is depicted in Figure 2.

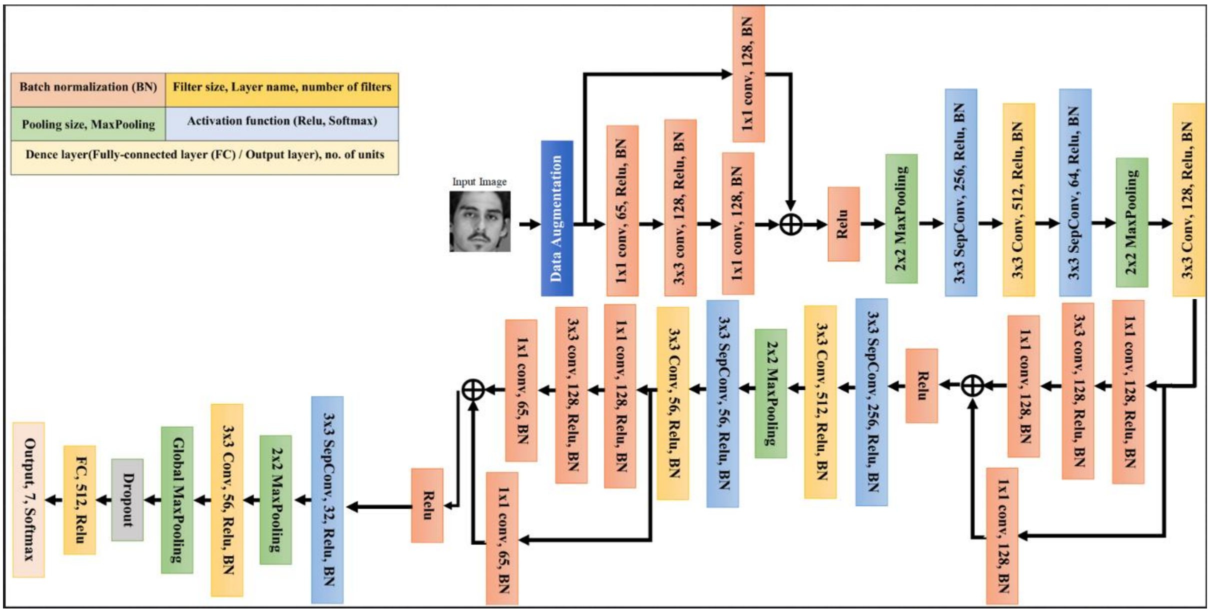

Figure 2 The MuWNet CNN model architecture

Data augmentation pre-processing was performed within the model to make these techniques active only during training. These approaches include horizontal flip, adjusting the width and height by $\pm 20^{\circ}$; zooming by $\pm 15^{\circ}$; rotation by $\pm 15^{\circ}$; and finally, normalizing the image pixel values by dividing them by 255. Figure 3 displays several examples of data augmentation on a random image.

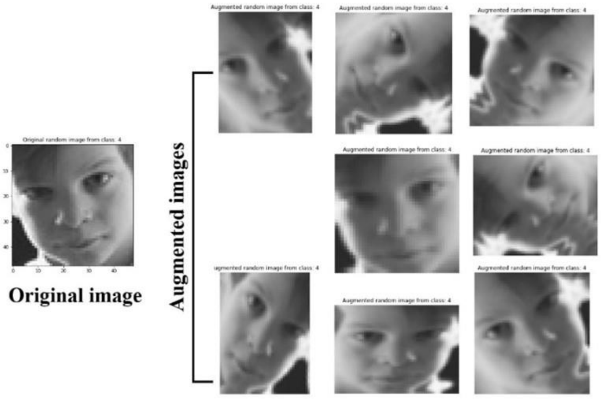
Original random image from class: 4
Original image
Augmented images
Augmented random image from class: 4

Figure 3 Examples of Augmented image.

3- EVALUATION

Therefore, the evaluation procedure demonstrates how well the model applied in this study performed. Along with the accuracy/loss plot, the study used performance indicators such as accuracy, recall, precision, and f1-score. Additionally, plotted accuracy versus loss.

IV. ENVIRONMENT EXPERIMENTAL SETTINGS

At the beginning of this research project, the data processing was done with Python. In addition, this investigation made use of Google CoLab as the foundational environment because it enables us to access a robust graphics processing unit (GPU) and makes it straightforward for us to construct and validate the proposed model. Specifications for Google CoLab: GPU Tesla P100-PCIE-16 GB, Random Access Memory (RAM) 12 GB, and Hard Disk Drive (HHD) 124 GB.

V. RESULTS & DISCUSSION

1- Hyperparameters Selection

Based on previous research and knowledge about the hyperparameters, this research employed a specific set of hyperparameters and established the range of values for their examination. In (Vulpe-Grigorasi & Grigore 2021) authors applied the dropout hyperparameter to demonstrate how different dropout values could alter the model's performance. In addition, researchers (Agrawal & Mittal 2019) utilized a variety of Optimizers to check their influence on the model accuracy compared to the ADAM optimizer, which is considered a default one. Moreover, the remaining hyperparameters, which are Image Size, Batch-Size, and Fully connected layer, were chosen for this study to investigate their impact on the model's performance. Finally, distinct values within a defined range were produced for each parameter. Table 1 summarizes the ranges, supported by the epochs' number, that were utilised to find the best parameter:

Table 1 Values of the hyperparameters

Hyperparameter	Value Range	No. of epochs
Image Size	[48, 64, 128, 150]	25
Optimizers	['Adam', 'SGD', 'RMSprop']	25
Batch_Size	[32, 64, 128]	20
Fully connected layer	[32, 64, 128, 512, 1028]	25
Dropout	[0.1, 0.2, 0.3, 0.4, 0.5]	20

After uploading the FER2013 dataset to Google Drive, the experiments ran on Google CoLab. Each best hyperparameter was determined independently, which is done by running an experiment for each hyperparameter, beginning with image size, and ending with dropout.

The results were acquired by tweaking the hyperparameters of the proposed CNN-based model classifier. The evaluation was performed by summing the validation accuracy for each epoch and dividing the entire sum by the epochs' number specified for each parameter. The average validation for both accuracy and loss for each parameter using the MuWNet model network are shown in Table 2.

Table 2 The average validation accuracy and average validation loss for MuWNet

Hyperparameter	No. of epochs	Values	Average validation accuracy	Average validation loss
Image Size	25	48×48	42.70%	147.15
		64×64	44.03%	143.44
		128×128	50.86%	127.77
		150×150	51.92%	125.05
Optimizers	25	'SGD'	42.69%	147.99
		'RMSprop'	48.26%	139.41
		'Adam'	51.92%	126.46
Batch_Size	20	32	50.15%	131.23
		64	49.66%	132.11
		128	47.66%	137.07
Fully connected layer	25	32	51.16%	126.56
		64	51.26%	126.97
		128	52.23%	124.87
		512	53.49%	121.25
		1028	53.33%	121.78
Dropout	20	0.1	48.34%	136.28
		0.2	49.46%	133.34
		0.3	48.36%	135.60
		0.4	46.76%	138.18
		0.5	45.96%	139.77

Table 3 shows the model's final parameters after the tuning trials for the hyperparameters were done.

Table 3 The parameters selected for the MuWNet model.

Image Size	Optimizers	Batch_Size	Fully connected layer	Dropout
150×150	'Adam'	32	512	0.2

Furthermore, in this study, the learning rate was adjusted using the ReduceLROnPlateau offered by Keras callbacks. This callback aims to fine-tune model weights by slowing down the rate of learning when the model performance stops improving.

2- MUWNET MODEL RESULTS ON THE FER2013 DATASET

The proposed model was used to conduct two experiments in order to acquire optimal outcomes. In the first experiment, the MuWNet model was trained using training data. In addition, the learning rate began at 0.001 and was changed throughout training using ReduceLROnPlateau, as described in the preceding chapter. This strategy was used to fine-tune model weights and overcome the issue of overfitting. The model was initially trained for 150 epochs. ReduceLROnPlateau decreased the learning rate from 0.001 to 0.0001 at epoch 135 during training, as shown in Figure 4. After 150 epochs, the validation accuracy (PublicTest) was 68.10% and the test data accuracy (PrivateTest) was 70.41%.

To improve the findings, the MuWNet model was trained for an additional 50 iterations. This demonstrated a shift in the learning rate from 0.0001 to 0.00001 at epoch 187, and at the end of these epochs, the model enhanced the outcomes by 68.43% for the validation data and by 70.52% for the test data.

In addition, looking at Figure 4, the model showed overfitting behavior with a training loss of 0.5612 and a validation loss of 0.9696 at epoch 200. As a result, 50 additional epochs were executed to surpass the training results and attempt to minimize the loss values for both training and validation to achieve the optimal fit. Finally, the results were improved after training the model for 250 epochs, with the model's accuracy on test data being 70.58% and a learning rate of 0.00001.

The accuracy/loss plots for this experiment are illustrated in Figures 4 and 5. The accuracy/loss plots showed that the learning rate changed by 0.0001 and 0.00001 at epochs 135 and 187, respectively. After epoch 187, the model's accuracy and loss almost stopped changing because it could not eliminate the problem of overfitting. That made the model too complicated to learn more from the same set of data. Table 4 presents the performance metrics for the first experiment.

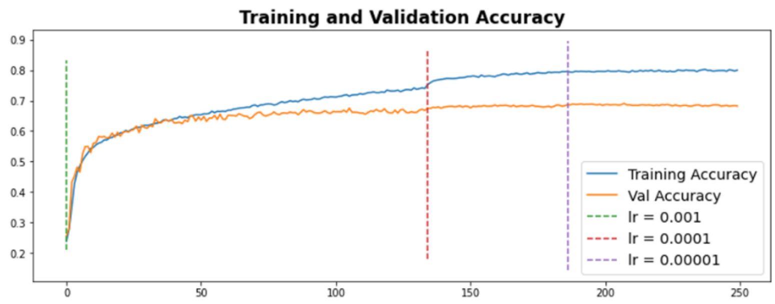

Figure 4 First experiment accuracy plot for MuWNet model on FER2013 dataset

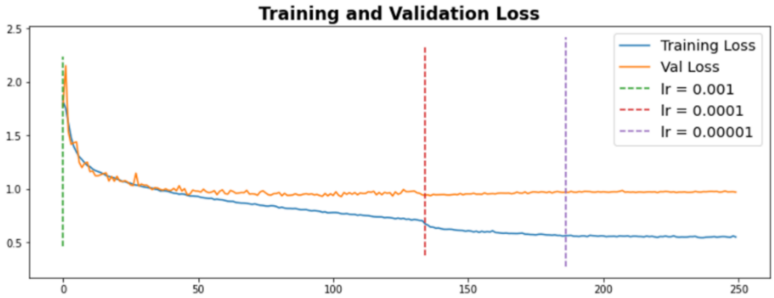

Figure 5 First experiment loss plot for MuWNet model on FER2013 dataset

Table 4 The parameters selected for the MuWNet model.

Accuracy	Recall	Precision	F1-Score
70.58%	69.61%	69.76%	69.65%

In the second experiment, training and validation data were combined into a single training dataset. The MuWNet model was then trained for 150 epochs. The model's accuracy on validation data was 79.86%, and on test data it was 68.01%.

The accuracy of the test data was less than what was achieved by the same model in the first experiment, and the learning rate's value did not alter during training; therefore, the value of the learning rate was modified manually from 0.001 to 0.0001 to train the model for an extra 50 epochs. After completing the additional 50 epochs, the model obtained an accuracy of 85.71% on validation data and 70.72% on test data, respectively. In the second experiment, the MuWNet model achieved loss values of 0.5412 for training and 0.4110 for validation at epoch 200, as shown in Figure 6 and 7.

Fifty training epochs were added to the model's total epochs to enhance the performance and reduce the training and validation loss values. By the time the training was over, the model had barely changed, and the test data accuracy had stayed at 70.72%. At the same time, the training and validation loss values were reduced to 0.5076 and 0.3738, respectively, which indicates that the model is learning and requires extra training epochs to alter its accuracy value. Figures 6 and 7 reveal this experiment's accuracy and loss plots. Furthermore, Table 5 presents the performance metrics for the second experiment.

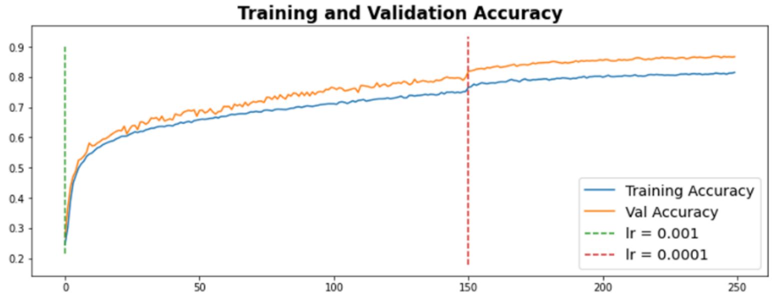

Figure 6 Second experiment accuracy plot for MuWNet model on FER2013 dataset

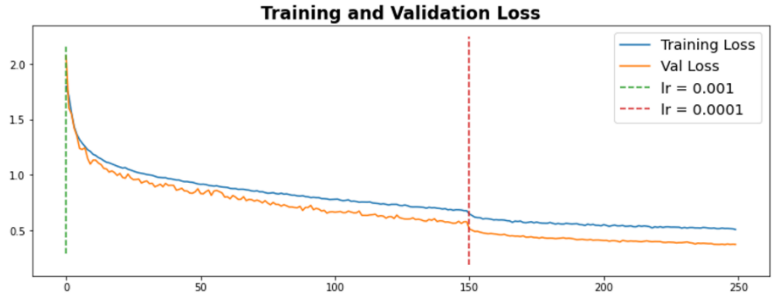

Figure 7 Second experiment loss plot for MuWNet model on FER2013 dataset

Table 5 The parameters selected for the MuWNet model.

Accuracy	Recall	Precision	F1-Score
70.72%	70.34%	69.61%	69.93%

In the second experiment, the combined dataset from the training and validation datasets exposed the model to more instances. In addition, feeding this dataset to the model reduced the gap between training and validation accuracy and loss values during training, which helped the MuWNet model perform significantly better at classifying emotions than it did in the first experiment, with an increase of 0.14% in the accuracy value. Figure 8 demonstrates the variance in the classification of emotions between the two MuWNet model experiments.

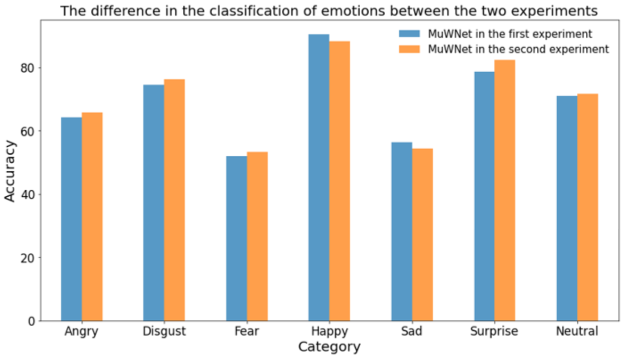

Figure 8 The difference in the classification of emotions between the two experiments

Table 6 shows how the model from the first experiment was compared to models from other studies that also used the FER2013 dataset.

Table 6 The comparison between the proposed model and other studies

Author(s)	Accuracy on FER2013 Dataset	No. of parameters
(Yanling Gan et al. 2019)	73.73%	—
(Khaireddin & Chen, 2021)	73.28%	—
(Abbassi et al. 2020)	72.806%	—
(Vulpe-Grigorasi & Grigore 2021)	72.16%	5.17M
(Sang et al. 2017)	71.9%	4.19M
(Shao & Qian 2019)	71.14%	7.12M
The proposed MuWNet model	**70.58%**	**3.1M**
(Onyema et al. 2021)	70%	—
(Cotter 2020)	67.96%	75,079
(Agrawal & Mittal 2019)	65.77%	0.93M
(Yijun Gan 2018)	64.24%	—
(Bhandari & Pal, 2021)	63.7%	9.9M
(Gaddam et al. 2022)	55.6%	—

the MuWNet model resulting from the second experiment was applied in real-time, and some real-time examples are presented in Figure 9.

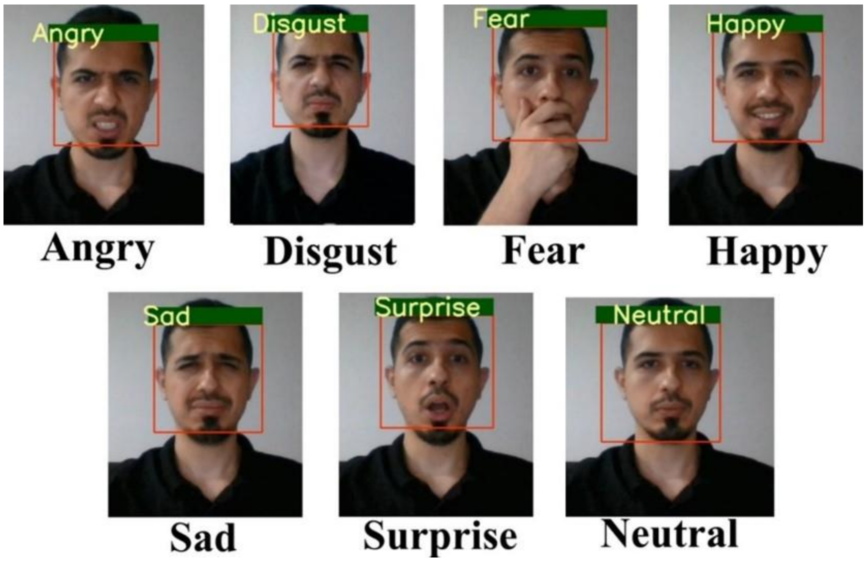

Figure 9 Some samples of real-time use of the MuWNet model

VI. CONCLUSION

This research aims to investigate the techniques and features used for facial emotion recognition in images using a novel DL model called MuWNet for classifying seven distinct facial emotions. In addition, comparing the results obtained with previous studies confirmed that the model can accurately organize emotions. The categorization of the FER2013 dataset served as the model challenge for this study. Finally, the MuWNet model was evaluated in real-time to capture human emotions.

This study was aided by the adaptation of several contemporary DL techniques and the usage of several layer types, such as convolution layers, separable convolution layers, and residual blocks. Moreover, employing hyperparameter adjustment was a good technique to boost the performance of various models, according to the findings of related works. Different layers, such as the BN Layer and Dropout Layer, were utilized in this investigation due to their usefulness in stabilizing training outcomes and minimizing overfitting, as reported by other works.

Accordingly, the suggested model contributed to the evolution of a face-expression classification system based on a DNN model. In addition, applying several hyperparameters and analyzing their influence on the suggested model.

Two experiments were conducted on the MuWNet model, and the findings show that the classification accuracy of the second experiment was 70.72%, with an increase of 0.14% over the first.

In terms of future work related to this study, such recommendations are illustrated by using different Datasets like AffectNet and the Extended Cohn-Kanade Dataset (CK+) can be an extension of this research. Concentrate on exploring the hyperparameter space using various strategies, such as grid search. Due to the disparity in performance between classes, employing various kernel_initializer classes may yield fruitful results. Similarly, using a different kernel_regularizer, such as L1 or L2, could improve the model's performance. A facial recognition system can be set up with a simple game so that it can quickly and accurately choose the right emoticon for any facial expression. Resolving the illumination problem by incorporating speech processing as a second input alongside the image.

REFERENCE

Abbassi, N., Helaly, R., Hajjaji, M.A. & Mtibaa, A. 2020. A Deep Learning Facial Emotion Classification system: A VGGNet-19 based approach. *Proceedings - STA 2020: 2020*

20th International Conference on Sciences and Techniques of Automatic Control and Computer Engineering 271–276. [11 July 2022].

Agrawal, A. & Mittal, N. 2019. Using CNN for facial expression recognition: a study of the effects of kernel size and number of filters on accuracy. *The Visual Computer 2019 36:2 36*(2): 405–412. https://link.springer.com/article/10.1007/s00371-019-01630-9 [11 July 2022].

Bhandari, A. & Pal, N.R. 2021. Can edges help convolution neural networks in emotion recognition? *Neurocomputing 433*: 162–168. [11 July 2022].

Carrier, P.-L. & Courville, A. 2013. Challenges in Representation Learning: Facial Expression Recognition Challenge | Kaggle https://www.kaggle.com/c/challenges-in-representation-learning-facial-expression-recognition-challenge/data [10 July 2022].

Cotter, S.F. 2020. MobiExpressNet: A deep learning network for face expression recognition on smart phones. *Digest of Technical Papers - IEEE International Conference on Consumer Electronics 2020-January* [6 July 2022].

Gaddam, D.K.R., Ansari, M.D., Vuppala, S., Gunjan, V.K. & Sati, M.M. 2022. Human Facial Emotion Detection Using Deep Learning. *Lecture Notes in Electrical Engineering 783*: 1417–1427. https://link.springer.com/chapter/10.1007/978-981-16-3690-5_136 [6 July 2022].

Gan, Yanling, Chen, J. & Xu, L. 2019. Facial expression recognition boosted by soft label with a diverse ensemble. *Pattern Recognition Letters 125*: 105–112. [11 July 2022].

Gan, Yijun. 2018. Facial expression recognition using convolutional neural network. *ACM International Conference Proceeding Series* [11 July 2022].

He, K., Zhang, X., Ren, S. & Sun, J. 2016. Deep residual learning for image recognition. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition 2016-December*: 770–778. [11 July 2022].

Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M. & Adam, H. 2017. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications https://arxiv.org/abs/1704.04861v1 [4 October 2022].

Kaviya, P. & Arumugaprakash, T. 2020. Group Facial Emotion Analysis System Using Convolutional Neural Network. *Proceedings of the 4th International Conference on Trends in Electronics and Informatics, ICOEI 2020* 643–647. [6 July 2022].

Khaireddin, Y. & Chen, Z. 2021. Facial Emotion Recognition: State of the Art Performance on FER2013 https://arxiv.org/abs/2105.03588 [5 July 2022].

Onyema, E.M., Shukla, P.K., Dalal, S., Mathur, M.N., Zakariah, M. & Tiwari, B. 2021. Enhancement of Patient Facial Recognition through Deep Learning Algorithm: ConvNet. *Journal of Healthcare Engineering 2021* [11 July 2022].

Pathar, R., Adivarekar, A., Mishra, A. & Deshmukh, A. 2019. Human Emotion Recognition using Convolutional Neural Network in Real Time. *Proceedings of 1st International*

Conference on Innovations in Information and Communication Technology, ICIICT 2019 [6 July 2022].

Sang, D.V., van Dat, N. & Thuan, D.P. 2017. Facial expression recognition using deep convolutional neural networks. *Proceedings - 2017 9th International Conference on Knowledge and Systems Engineering, KSE 2017 2017-January*: 130–135. [11 July 2022].

Shao, J. & Qian, Y. 2019. Three convolutional neural network models for facial expression recognition in the wild. *Neurocomputing 355*: 82–92. [11 July 2022].

Simonyan, K. & Zisserman, A. 2014. Very Deep Convolutional Networks for Large-Scale Image Recognition. *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings* https://arxiv.org/abs/1409.1556v6 [4 October 2022].

Vulpe-Grigorasi, A. & Grigore, O. 2021. Convolutional Neural Network Hyperparameters optimization for Facial Emotion Recognition. *12th International Symposium on Advanced Topics in Electrical Engineering, ATEE 2021* [11 July 2022].