Revue d'Intelligence Artificielle
Vol. 38, No. 4, August, 2024, pp. 1201-1209
Journal homepage: http://iieta.org/journals/ria

Facial Expression Recognition Using Deep Learning and Neural Embeddings

Muhamad Arief Liman*, Gede Putra Kusuma

Computer Science Department, BINUS Graduate Program - Master of Computer Science, Bina Nusantara University, Jakarta 11480, Indonesia

Corresponding Author Email: muhamad.liman@binus.ac.id

https://doi.org/10.18280/ria.380414

Received: 16 November 2023
Revised: 20 February 2024
Accepted: 26 April 2024
Available online: 23 August 2024

Keywords:
Facial Expression Recognition, deep learning, neural embedding, triplet loss, machine learning classifier

ABSTRACT

This study investigates Facial Expression Recognition (FER) as essential for understanding human emotions conveyed through facial expressions, involving face detection, facial expression detection, and classification. Recent advancements in deep learning have significantly enhanced FER accuracy, exemplified by combining Visual Geometry Group (VGG) and U-Net segmentation layers, achieving a remarkable 75.97% accuracy. Building upon prior research on neural embeddings, this study explores their application in improving FER models, focusing on basic models like VGG-19 and employing triplet loss. Extracted features are classified using various methods such as Support Vector Machine, XGBoost, Random Forest, and Artificial Neural Networks, with evaluation metrics including accuracy, precision, recall, and F1 Score. Findings indicate that modifications to the VGG19 classifier improve accuracy, with XGBoost attaining the highest accuracy of 65.70%. However, integrating triplet loss does not yield significant improvement, recording a highest accuracy of 65.30% when combined with the XGBoost model. These results suggest potential limitations, such as incorrect distance calculation methods and dataset imbalance, which need addressing for enhancing model efficacy and real-world applicability. Therefore, future research should focus on refining distance calculation techniques and ensuring dataset balance.

1. INTRODUCTION

Human beings are inherently social creatures, and one of the fundamental ways emotions are expressed is through interaction. Specifically, emotions are indicated through facial expressions. In 1967, a study was conducted by Marechal et al. [1] which revealed that a substantial 55% of emotional communication is visually conveyed through facial expressions. The crucial role played by facial expressions in our social lives is underscored by this finding. Furthermore, facial expressions have been categorized into two distinct groups, basic emotions and compound emotions, emphasizing the complexity of human emotional expression [2]. Facial Expression Recognition (FER) has emerged as a field with considerable potential for various applications across domains such as education, healthcare, security, and more [3].

In the Facial Expression Recognition process, three main stages are involved: face detection, facial expression detection, and classification. Research conducted by Tian et al. [4] demonstrates on Figure 1, that there are three approaches to performing Facial Expression Recognition. These approaches include Face Acquisition, Facial Data Extraction and Facial Expression Recognition. However, the recognition of facial expressions is by no means an uncomplicated task. Challenges such as variations in head pose, age, gender, backgrounds, the presence of accessories, and even underlying health conditions are confronted in the process. In recent years, the landscape of FER research has been transformed by deep learning, providing a more robust approach that eliminates the need for manually crafted rules. Convolutional neural networks (CNNs), particularly renowned for their object-detection capabilities, are frequently employed.

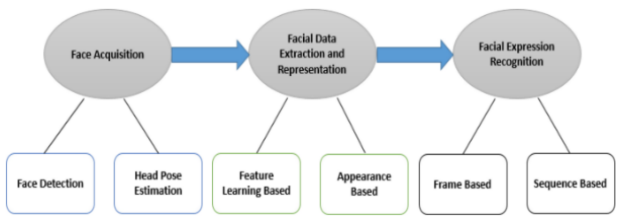


Figure 1. Facial Expression Recognition process

Beyond CNNs, several other algorithms, such as VGG [5], ResNet [6], MobileNet [7], and MobileNetV2 [8], have gained prominence, making FER more accessible and accurate. In the realm of facial expression recognition (FER) research, one noteworthy study conducted by Vignesh et al. [9]. In 2023 introduced a model that combined the Visual Geometry Group (VGG) layer with the U-Net segmentation layer. This model was meticulously trained using the FER2013 dataset, which comprises seven distinct emotional classes. The integration of the VGG-19 and U-Net segmentation layers was a strategic move aimed at enhancing the significance of critical features extracted from feature maps. This approach enabled the model to exercise control over information redundancy within the

VGG layers. U-Net effectively reconstructs the original image using the acquired feature map. The outcome of this research was particularly noteworthy, achieving the second-highest accuracy among benchmarks established by paperswithcode, with an impressive accuracy rate of 75.97%.

In a complementary study by Santoso and Kusuma [10] in 2022, several hybrid models were developed, including combinations of VGGNet with SpinalNet, Vision Transformer with SpinalNet, and EfficientNetV2 with SpinalNet. This research introduced innovative architectural combinations, modifying classifiers from various models by incorporating SpinalNet. Notably, the most successful model emerged from the fusion of VGGNet and SpinalNet, boasting an accuracy rate of 74.45%. These research endeavors collectively demonstrate that these models have consistently achieved high accuracy levels in the domain of facial expression recognition.

From previous research, in this paper the possibility of enhancing existing FER models through the utilization of neural embeddings is explored. Neural embeddings, which transform data from a higher-dimensional space to a lower-dimensional one, play a critical role in the performance of FER models. These loss functions aim to minimize the distance between embeddings, including the application of contrastive loss or triplet loss. Base models like VGG-19 are leveraged as feature extractors, and triplet loss, combined is applied to train models that generate more informative features as a loss function. These features are subsequently classified using various classifiers, such as Support Vector Machine [11], XGBoost [12], Random Forest and artificial neural networks (ANN) [13]. Within the realm of FER research, the potential for significant advancements in the recognition of human emotions through facial expressions is seen as resulting from the fusion of deep learning and neural embeddings.

2. RELATED WORKS

Vignesh et al. [9] conducted a model by combining the Visual Geometry Group layer with the U-Net segmentation layer, trained on the FER2013 dataset with seven distinct classes. Integrating these layers significantly impacted critical features, enabling control over information flow redundancy within the VGG layers. The U-Net architecture, with its U-shaped design incorporating downsampling and upsampling components, effectively restored feature mapping to the original image post-downsampling. This research achieved the second-highest accuracy among benchmarks on paperswithcode, reaching an impressive 75.97% accuracy rate.

Undertook research involving several combined models, including VGGNet with SpinalNet, Vision Transformer with SpinalNet, and EfficientNetV2 with SpinalNet [10]. The authors proposed architectures that modified classifiers from these models using SpinalNet. For instance, in the case of VGGNet, the architecture featured 4 Convolution Blocks combined with SpinalNet's 4 Spinal Layers. Similarly, within the Vision Transformer framework, three primary processes were executed: the Patch and Position Embedding Layer, the Transformer Encoder Layer, and the MLP Head. After these processes, the resulting embeddings were merged, with SpinalNet serving as the classifier. The amalgamation of VGGNet and SpinalNet yielded the best performance, achieving an accuracy of 74.45%.

The study introduced the LHC-Net model, trained on the FER2013 dataset, achieving an accuracy of 74.42% [14]. This model, resembling ResNet34 in architecture, integrates local head channel fusion. LHC-Net is founded on two main principles. Firstly, it suggests that in computer vision, utilizing the self-attention paradigm based on channels, rather than spatial attention, is most effective. Secondly, it asserts that a local approach shows potential in overcoming the limitations of convolution compared to global attention.

Formulated the VGGNet model, which was trained on the FER2013 dataset, a compilation of facial expressions for recognition [15]. The dataset comprises seven classes: angry, disgusted, fearful, happy, sad, surprised, and neutral. This model achieved an accuracy of 73.28%, featuring an architecture comprising four convolutional layers, max pooling, and three fully connected layers, all employing ReLU activation functions.

Punuri et al. [16] introduced the Efficient Net-XGBoost model, which merges the EfficientNet model as a feature layer with XGBoost serving as a classifier. This model incorporates the Efficient Net architecture by hooking into pooling layers to obtain feature maps used in the XGBoost classifier. The research employed several facial expression datasets, including CK+, KDEF, JAFFE, and FER2013. The outcomes were impressive, with accuracy rates of 100%, 99%, and 98% achieved on CK+, KDEF, and JAFFE, respectively. For the FER2013 dataset, an accuracy rate of 72.54% was attained.

Fard and Mahoor [17] introduced a method utilizing adaptive correlation-based loss, directing the network to produce embedded feature vectors with elevated correlation for samples within the same class and reduced correlation for samples between classes. Ad-Corre comprises three elements: a feature discriminator instructing the network to generate highly correlated embedded feature vectors for samples of the same class and less correlated ones for different classes, an average discriminator guiding the network to ensure the dissimilarity of average embedded feature vectors across different classes, and a functional embedding discriminator penalizing the network for generating diverse embedded feature vectors.

Vulpe-Grigorasi and Grigore [18] developed a CNN model with hyperparameter tuning, achieving an accuracy of 72.16%. Optimization was conducted to obtain the best hyperparameters, utilizing the Random Search Algorithm. The results revealed a learning rate of 0.001, a batch size of 128, and a 3×3 kernel with ReLU activation function, complemented by two fully connected layers featuring 256 neurons in the first layer and 7 neurons in the second layer.

The study developed a Convolutional Neural Network (CNN) model using transfer learning, comprising three convolution layers, four pooling layers, three fully connected layers, and a classification layer, with an input size of 48×48 [19]. The research began with face detection and underwent preprocessing, including augmentation, rotation/flip, and normalization. Initial weights and biases were obtained from Facial Expression Recognition (FER) data, and transfer learning was conducted using the CF+ and JAFFE datasets to enhance performance across various tasks. The model achieved an accuracy of 71.45% on the FER2013 dataset.

Huo et al. [20] employed MobileNetV2 with a Gaussian filter and Canny operator, combined with a Softmax classifier. The Gaussian filter, applied using the Canny operator, was utilized to eliminate image noise and merge two original pixel feature maps into a three-channel image. The result of this Gaussian filter process was trained using the MobileNetV2

network, which employed ReLU6 as a nonlinear activation function and concluded with a Softmax classifier. The experiments yielded an accuracy of 70.76% on the FER2013 dataset and an impressive 97.92% on the CK+ dataset.

The Attentional Convolution Network was introduced, integrating a CNN model with an attention mechanism [21]. This model comprised four convolution layers, each followed by a pooling layer with ReLU activation functions and dropout layers. Additionally, spatial transformers were employed, featuring two convolution layers with pooling and ReLU activation layers. These transformers enhanced focus on relevant image patches by estimating sample correlations. Training utilized the SGD optimizer and cross-entropy loss function, resulting in an accuracy of 70.02% on the FER2013 dataset. Due to the satisfactory performance of the VGG19 model and its adaptable layer structure, it was selected for use in this research.

From previous research, several insights have emerged. It has been observed that altering the loss function and substituting the model can enhance model performance, with accuracy serving as the benchmark. In this research, the proposed approach involves replacing the loss function with triplet loss, a neural embeddings method, and substituting the classifier with a machine learning model. It is anticipated that these adjustments will enhance the performance of the VGG-19 model.

3. METHODOLOGY

3.1 Dataset

The dataset used for this paper is FER2013, which is a challenge dataset available on Kaggle. This dataset was created by Goodfellow et al. [22] and comprises a total of 35,887 images and pixels, divided into 28,709 training data and 7,178 test data. FER2013 consists of 7 classes: anger, disgust, fear, happiness, sadness, surprise, and neutral. The data size is 48×48 gray images, with data distribution as presented in Table 1. and the graph in Figure 2.

Table 1. Dataset FER2013

Category name	Distribution
Angry	4953
Fear	5121
Sad	6077
Neutral	6198
Happy	8989
Surprise	4002
Disgust	547
Total	**35887**

Figure 2. FER2013 examples images

FER2013 has gained popularity among researchers and machine learning practitioners due to its accessibility and the substantial amount of data it offers. It has been widely utilized in various machine learning papers and competitions for the task of facial emotion recognition. This dataset proves valuable for building emotion recognition systems that can be applied in practical scenarios, such as understanding emotions in video conferencing and gaming applications. However, there are limitations to this dataset, specifically regarding the distribution of data, which is imbalanced. For instance, the 'disgust' class contains only 547 images.

3.2 Model development

At the forefront of facial expression recognition research, significant contributions have been made by numerous studies. For instance, a model that combines the Visual Geometry Group (VGG) layer with a U-Net segmentation layer was developed by Vignesh et al. [9]. This model was trained using the FER2013 dataset, which comprises 7 distinct emotion classes. The fusion of VGG-19 with the U-Net Segmentation layer yields significant improvements in extracting crucial features from the feature map. This integration effectively manages the flow of redundant information within the VGG layer. VGG-19 is renowned for its simplicity and efficacy, featuring 19 layers encompassing convolutional layers with compact 3×3 filters and max-pooling layers, followed by fully connected layers.

The opportunity for model development from the utilized architectures is identified. Feature values generated by each model are extracted using hooks at the global average pool layer. To maximize these feature values, the models are trained using triplet loss, a deep learning loss function for metric learning [23]. Triplet loss involves three input examples: anchor, positive, and negative. Anchor and positive are instances belonging to the same class or category, while negative belongs to a different class or category. The goal of triplet loss is to minimize the distance between anchor and positive, given their shared class, and maximize the distance between anchor and negative, due to their differing classes, as illustrated in Figure 3.

The triplet loss architecture is utilized to handle distributed embedding by accounting for both similarity and dissimilarity. This triplet loss serves as a beneficial loss function to improve feature generation in VGG-19. The mathematical operation for calculating triplet loss is depicted in Eq. (1):

$$L_{(a,p,n)} = \max\left(0, D_{(a,p)} - D_{(a,n)} + \text{margin}\right) \quad (1)$$

Here, margin is determined by hyperparameters to define the distance between positive and negative inputs, while D represents the learned vector distance. Triplet selection, the process of selecting anchor, positive, and negative inputs for metric distance calculation, aims to provide effective model information and convergence.

Several methods for triplet selection exist, including Random Selection (inefficient due to random triplet choice), Semi-Hard Triplet Mining (choosing a negative closer to the anchor than the positive), and Hard Triplet Mining (selecting the negative example with the closest distance, extending Semi-Hard Mining but increasing model complexity). Triplet loss employs distance calculations such as Euclidean distance, Manhattan Distance, or Cosine Similarity between anchor and positive or negative samples.

Once the model is trained using the combined triplet loss method, features are extracted through hooks at the global average pooling layer in VGG-19. These feature values undergo preprocessing for use in classifier models like SVM, XGBoost, and Artificial Neural Networks (ANN). Finally, model performance is evaluated to compare and consider both trained models. As shown in Figure 4.

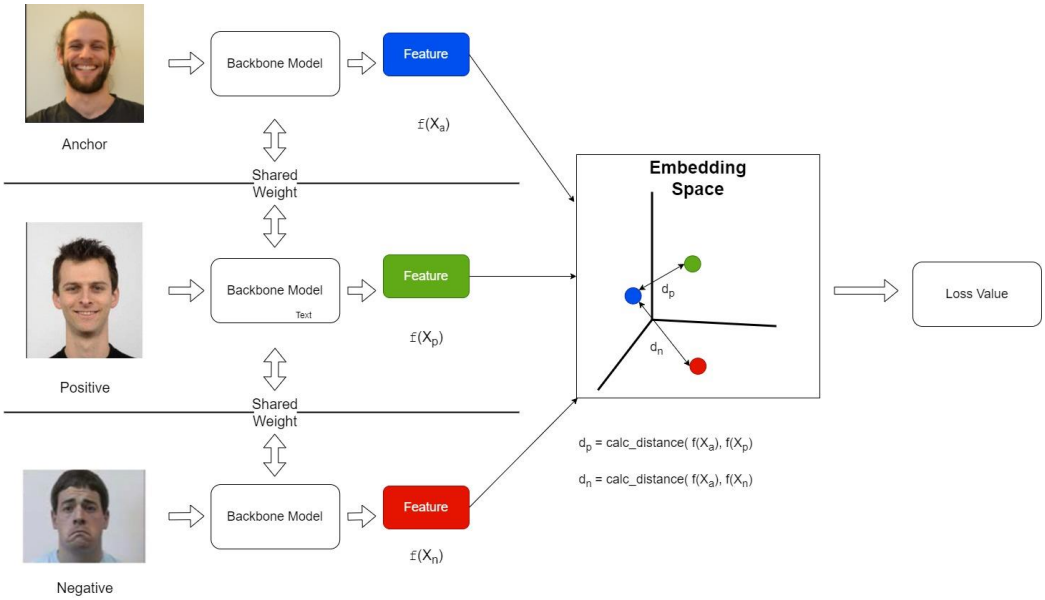


Figure 3. Illustration of process on triplet loss

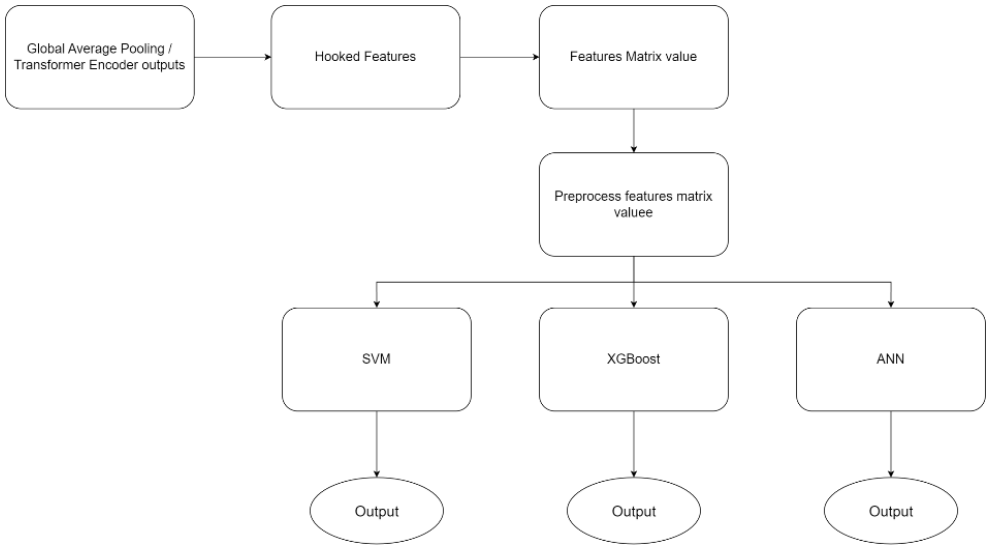


Figure 4. Features extraction with model classifier

3.3 Experimental design

The dataset used in this research is the Open Dataset FER2013, which consists of 35,887 images sized at 48×48 pixels. This dataset has been divided into two subsets, namely, training and testing data. The training dataset comprises a total of 28,709 data points, while the testing dataset contains 7,178 data points. The testing data is further categorized into two segments, namely, public test and private test. Public test data can be used for validation and includes 3,589 data points. All of these data points are categorized into seven classes based on the corresponding emotions, and the distribution of data can be observed in Table 1.

During the training phase, the model selected is a pre-trained model, which has been previously trained on a large dataset such as ImageNet. This trained model undergoes transfer learning to adjust its weights and biases using the FER2013 dataset. The pre-trained model is trained using triplet loss and cross-entropy loss. Model training covered 75 epochs, combining early stopping with a patience of 10. Additionally, the optimization process used SGD to update model weights and biases. A summary of the Hyperparamter setup for the VGG19 baseline model is shown in Table 2. After successfully training the model using custom triplet loss, Torch is used to obtain feature maps via hook. Model evaluation entails the utilization of various performance metrics that are readily available. Subsequently, the model will undergo training using a training dataset comprising 28,709 samples, along with validation data consisting of 3,589 samples and test data comprising an equal number of 3,589 samples. The environment used in this research is Kaggle with GPU T4×2.

Table 2. Summary of hyperparameters setup for VGG19 baseline model

Parameters	VGG19 Baseline Model
Pretrain	ImageNet1K
Classifier	1 Linear Layer with Softmax
Epochs	75
Optimizer	SGD with 0.001 Learning Rate and 0.9 Momentum
Scheduler	ReduceOnPlateau - Patience 3 Mode min
Total Params	20,027,975

3.4 Performance metrics

Performance measurement will be conducted on each trained model in Figure 5. Confusion matrices, as represented in Table 3, will be employed for evaluation. Evaluation metrics will be used to calculate accuracy, precision, recall, and F1-Score for each class. Additionally, measurements will be made for the average values of accuracy, precision, recall, and F1-Score.

Class		Predicted							
		1	2	3	4	5	6	7	Total
Actual	1	T11	F12	F13	F14	F15	F16	F17	C1
	2	F21	T22	F23	F24	F25	F26	F27	C2
	3	F31	F32	T33	F34	F35	F36	F37	C3
	4	F41	F42	F43	T44	F45	F46	F47	C4
	5	F51	F52	F53	F54	T55	F56	F57	C5
	6	F61	F62	F63	F64	F65	T66	F67	C6
	7	F71	F72	F73	F74	F75	F76	T77	C7
Total		C1'	C2'	C3'	C4'	C5'	C6'	C7'	All

Figure 5. Sample of confusion matrix with 7 class

$$Accuracy = \frac{(T11 + T22 + T23 + T24 + T25 + T26 + T27)}{All} \quad (2)$$

$$Precision(i) = \frac{TP_{ii}}{C_i^{'}} \quad (3)$$

$$Recall(i) = \frac{TP_{ii}}{C_i} \quad (4)$$

$$F1 - Score(i) = 2 * \frac{Precision(i) * Recall(i)}{Precision(i) + Recall(i)} \quad (5)$$

$$Average\ Precision = \frac{\sum Precision(i)}{N} \quad (6)$$

$$Average\ Recall = \frac{\sum Recall(i)}{N} \quad (7)$$

$$Average\ F1 - Score = \frac{\sum F1 - Score(i)}{N} \quad (8)$$

Accuracy, a commonly employed evaluation metric, measures the proportion of correct predictions (including both positive and negative outcomes) relative to the entire available dataset. In contrast, precision signifies the ratio of true positive predictions to the total positive predictions made. Recall, on the other hand, expresses the ratio of true positive predictions to the total actual positive instances in the dataset. The F1-Score, serving as an average metric, is calculated by comparing precision and recall. Because the dataset used is imbalanced, therefore the performance metric used or valid is only F1-Score.

4. RESULT AND DISCUSSION

4.1 Training and validation result

The performance of the vgg19 baseline model, as indicated in Table 3, was disclosed through experiments on training and validation data. By tuning the pre-trained model with the FER2013 dataset, an accuracy of 86.39% for training data and 65.28% for validation data was achieved. The training phase was concluded with an early stop at epoch 43. Figure 6 illustrates the training and validation results for this experiment. Specifically, an improvement is observed in terms of accuracy and loss reduction in the baseline of this model.

By replacing the classifier in the baseline model with various classifiers such as ANN, SVM, Random Forest, and XGBoost, an improvement in accuracy can be achieved, as shown in Table 4. Where XGBoost attains the highest F1 Score with a value of 1.0 on training data.

Table 3. Summary of training and validation result using baseline model with cross entropy loss

Model	Data	Accuracy	Precision	Recall	F1-Score
Baseline	Training data	86.39	0.85	0.86	0.86
Baseline	Validation data	65.28	0.64	0.65	0.65

Table 4. Summary of training result using baseline model with cross entropy loss and various classifier

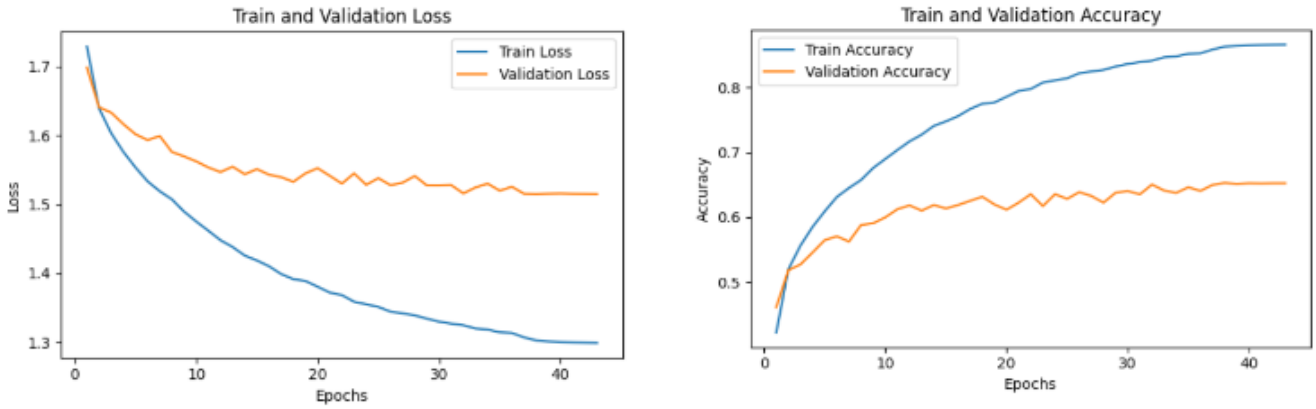
Method	Accuracy	Precision	Recall	F1-Score
Baseline + Cross Entropy + ANN	86.32	0.87	0.86	0.86
Baseline + Cross Entropy + SVM	73.93	0.74	0.74	0.74
Baseline + Cross Entropy + Random Forest	96.27	0.96	0.96	0.96
Baseline + Cross Entropy + XGBoost	99.84	1.0	1.0	1.0

Table 5. Summary of validation result using baseline model with cross entropy loss and various classifier

Method	Accuracy	Precision	Recall	F1-Score
Baseline + Cross Entropy + ANN	65.19	0.64	0.65	0.65
Baseline + Cross Entropy + SVM	55.44	0.57	0.55	0.55
Baseline + Cross Entropy + Random Forest	65.11	0.65	0.65	0.65
Baseline + Cross Entropy + XGBoost	99.74	1.0	1.0	1.0

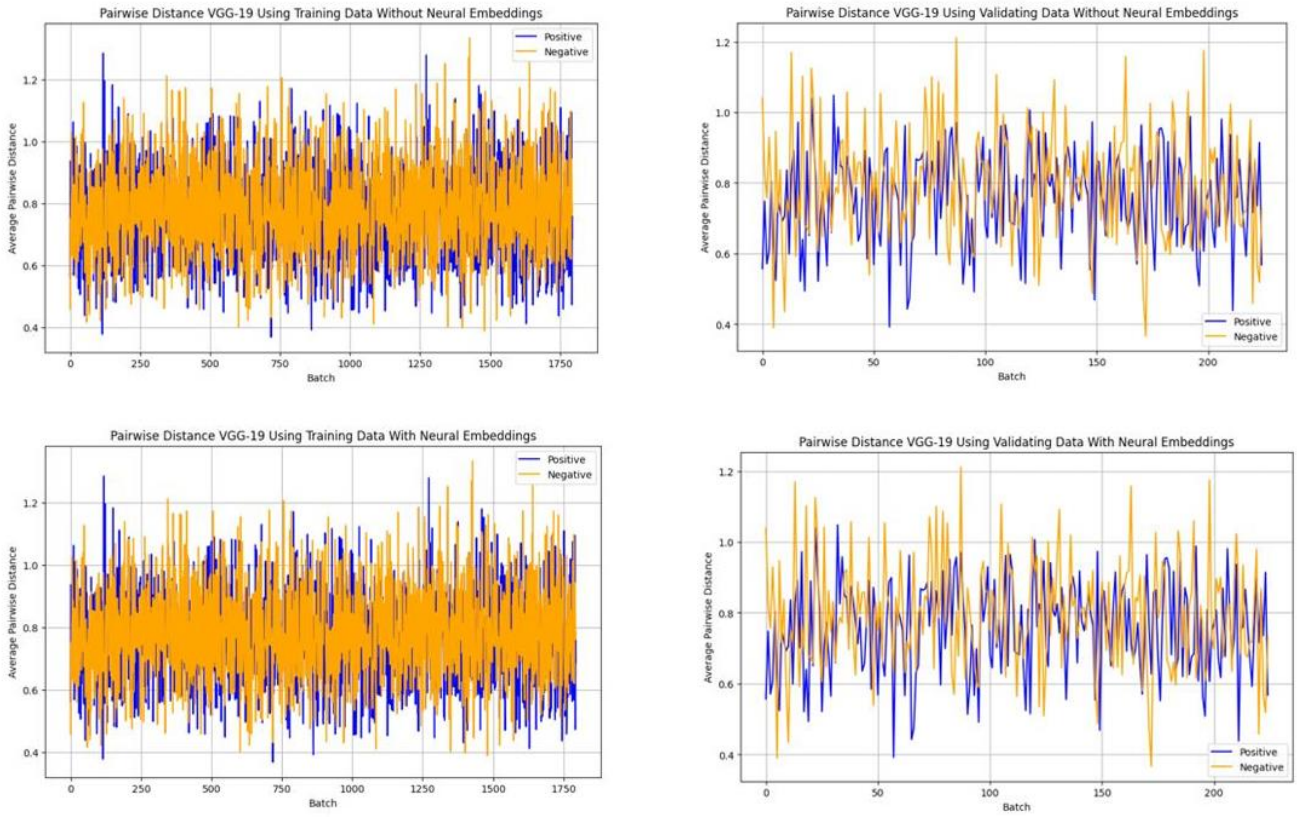
Table 6. Summary of training result using baseline model with triplet loss and various classifier

Method	Accuracy	Precision	Recall	F1-Score
Baseline + Triplet Loss + ANN	84.30	0.83	0.84	0.84
Baseline + Triplet Loss + SVM	90.89	0.91	0.91	0.91
Baseline + Triplet Loss + Random Forest	86.64	0.87	0.87	0.86
Baseline + Triplet Loss + XGBoost	99.83	1.0	1.0	1.0



(a) (b)

Figure 6. Train and validation plotting for baseline model: (a) Train and validation loss (b) Train and validation accuracy



(a) (b)

Figure 7. Train and validation pairwise distance for baseline model with and without using neural embeddings: (a) train data (b) validation data

Table 7. Summary of validation result using baseline model with triplet loss and various classifier

Method	Accuracy	Precision	Recall	F1-Score
Baseline + Triplet Loss + ANN	62.52	0.62	0.63	0.62
Baseline + Triplet Loss + SVM	62.38	0.65	0.62	0.63
Baseline + Triplet Loss + Random Forest	62.91	0.64	0.63	0.63
Baseline + Triplet Loss + XGBoost	99.77	1.0	1.0	1.0

Table 5 shows the performance results of using validation data by combining various classifiers with the baseline model, where XGBoost achieves an F1 Score of 1.0 compared to other classifier models.

By applying triplet loss to the VGG19 baseline model, the model will be trained without the presence of a classifier. Triplet loss will train the embedding in the baseline model by calculating the distance between input triplets. Figure 7 below depicts a plot where the embedding values produced by models trained using neural embeddings or triplet loss are no better than models trained without neural embeddings or triplet loss.

Table 6 shows the performance results of using training data on the baseline model that has applied triplet loss and various classifiers. In this context, the XGBoost classifier obtains the highest F1-Score with a value of 1.0.

Table 7 shows the performance results of using validation data by combining various classifiers with the baseline model and triplet loss, where XGBoost achieves an F1 Score of 1.0 compared to other classifier models. From the displayed results, it is indicated that some models may be overfit, as evidenced by significantly different values between training and validation.

4.2 Testing result

From the experiment on testing data, by changing the classifier in the baseline VGG19 model, an increase in accuracy is obtained compared to the initial classifier's usage. However, the application of Triplet Loss to the Baseline VGG19 model has not yielded better results than using Cross Entropy Loss. The Table 8 is the performance matrix of the proposed baseline model.

The Table 9 below displays the outcomes of the VGG19 model with the Model classifier. Using XGBoost, an improvement is observed, with an accuracy of 65.70% achieved on the test data. The best parameters were obtained after a total of 25 fits during Random Search, comprising Sub-Sample: 0.5, n_estimators: 500, min_child_weight: 2, max_depth: 10, learning_rate: 0.1, and colsample_bytree: 0.6.

The Table 10 below displays the results of the VGG19 model with Triplet Loss and the Model classifier. When using XGBoost, an improvement is observed, with an accuracy of 65.30% achieved on the test data. The best parameters were obtained after a total of 25 fits during Random Search, comprising Sub-Sample: 0.5, n_estimators: 500, min_child_weight: 2, max_depth: 10, learning rate: 0.1, and colsample_bytree: 0.6.

Table 8. Summary of testing result using baseline model with cross entropy loss

Method	Accuracy	Precision	Recall	F1-Score
Baseline + Cross Entropy	65.09	0.64	0.65	0.65

Table 9. Summary of testing result using baseline model with cross entropy loss and various classifier

Method	Accuracy	Precision	Recall	F1-Score
Baseline + Cross Entropy + ANN	65.05	0.65	0.65	0.65
Baseline + Cross Entropy + SVM	56.42	0.58	0.56	0.56
Baseline + Cross Entropy + Random Forest	64.45	0.65	0.64	0.64
Baseline + Cross Entropy + XGBoost	65.70	0.66	0.66	0.66

Table 10. Summary of testing result using baseline model with triplet loss and various classifier

Method	Accuracy	Precision	Recall	F1-Score
Baseline + Triplet Loss + ANN	63.40	0.63	0.63	0.63
Baseline + Triplet Loss + SVM	61.55	0.65	0.62	0.62
Baseline + Triplet Loss + Random Forest	64.28	0.65	0.64	0.64
Baseline + Triplet Loss + XGBoost	65.30	0.65	0.62	0.62

From the results presented above, changing the classifier using XGBoost can improve model performance. Where the classifier used is more complex and more accurate than a single classification layer. The use of triplet loss with Euclidean distance to calculate distance has not obtained maximum results and the dataset is also unbalanced.

5. CONCLUSIONS AND FUTURE WORKS

The experiments performed reveal that modifying the traditional VGG19 classifier, typically composed of one Linear layer with Softmax, leads to improved accuracy. This enhancement is attributed to the increased complexity introduced by employing Machine Learning Models such as SVM, Random Forest, XGBoost, and ANN, in contrast to the original VGG19 classifier. Specifically, replacing the classifier with XGBoost resulted in the highest accuracy of 65.70%, compared to the Baseline Model trained with an accuracy of 65.09%. However, the use of Triplet Loss has not yet produced more optimized features compared to Cross-Entropy. The highest accuracy achieved in this research was

65.70% when using Cross Entropy with classifier model XGBoost.

This research identified several shortcomings and areas requiring improvement in facial expression recognition. It underscores limitations in dataset quality, particularly the imbalance between classes, which leads to suboptimal model evaluation and performance. Additionally, the application of triplet loss does not yield superior model performance, partly due to limited implementation methods and the potential use of inappropriate Euclidean distances.

The first future works for this research involves enhancing classifier performance by exploring alternative models. It is important to note that trying several other models does not eliminate the possibility of improvement. The second area for future investigation entails modifying or amalgamating the FER2013 dataset with the FER dataset to achieve a more balanced dataset composition. Additionally, performing hyperparameter tuning to better align with the dataset characteristics represents the third avenue for future work. Moreover, exploring the potential of models trained on the FER dataset to recognize expressions beyond images, such as videos, constitutes the fourth area for future research. Finally, this research holds promise for advancing computer interaction in the future.

ACKNOWLEDGMENT

The authors would like to thank I. J. Goodfellow et al. from Université de Montréal for the FER2013 Dataset.

REFERENCES

[1] Marechal, C., Brazil, E., Kanaan, S., Nitsche, M., Masip, D., Ruiz-Hidalgo, J. (2019). Survey on AI-based multimodal methods for emotion detection. High-Performance Modelling and Simulation for Big Data Applications, 11400: 307-324. https://doi.org/10.1007/978-3-030-16272-6_11

[2] Du, S., Tao, Y., Martinez, A.M. (2014). Compound facial expressions of emotion. Proceedings of the National Academy of Sciences, 111(15): E1454-E1462. https://doi.org/10.1073/pnas.1322355111

[3] Konstantina, V., Anna, H., Thomas, Z. (2021). Facial Emotion Recognition. European Data Protection Supervisor TechDispatch. https://www.edps.europa.eu/data-protection/our-work/publications/techdispatch/techdispatch-12021-facial-emotion-recognition_en.

[4] Tian, Y., Kanade, T., Cohn, J.F. (2011). Facial expression recognition. Handbook of Face Recognition, pp. 487-519. https://doi.org/10.1007/978-0-85729-932-1_19

[5] Simonyan, K., Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. https://doi.org/10.48550/arXiv.1409.1556

[6] He, K., Zhang, X., Ren, S., Sun, J. (2016). Deep residual learning for image recognition. In 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, pp. 770-778. https://doi.org/10.1109/CVPR.2016.90

[7] 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://doi.org/10.48550/arXiv.1704.04861

[8] Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.C. (2018). Mobilenetv2: Inverted residuals and linear bottlenecks. In 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, pp. 4510-4520. https://doi.org/10.1109/CVPR.2018.00474

[9] Vignesh, S., Savithadevi, M., Sridevi, M., Sridhar, R. (2023). A novel facial emotion recognition model using segmentation VGG-19 architecture. International Journal of Information Technology, 15: 1777-1787 https://doi.org/10.1007/s41870-023-01184-z

[10] Santoso, B.E., Kusuma, G.P. (2022). Facial emotion recognition on FER2013 using VGGSpinalNet. Journal of Theoretical and Applied Information Technology, 100(7): 2088-2102. http://www.jatit.org/volumes/Vol100No7/10Vol100No7.pdf.

[11] Al Azies, H., Trishnanti, D., Mustikawati, E.P.H. (2019). Comparison of kernel support vector machine (SVM) in Classification of human development index (HDI). IPTEK Journal of Proceedings Series, 6. http://doi.org/10.12962/j23546026.y2019i6.6394

[12] Chen, T., Guestrin, C. (2016). Xgboost: A scalable tree boosting system. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 785-794. https://doi.org/10.1145/2939672.2939785

[13] Mishra, M., Srivastava, M. (2014t). A view of artificial neural network. In 2014 International Conference on Advances in Engineering Technology Research (ICAETR - 2014), pp. 1-3. https://doi.org/10.1109/ICAETR.2014.7012785

[14] Pecoraro, R., Basile, V., Bono, V., Gallo, S. (2021). Local multi-head channel self-attention for facial expression recognition. In 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp: 232-236. https://doi.org/10.48550/arXiv.2111.07224

[15] Khaireddin, Y., Chen, Z. (2021). Facial emotion recognition: State of the art performance on FER2013. https://doi.org/10.48550/arXiv.2105.03588.

[16] Punuri, S.B., Kapse, A.R., Chincholikar, S.S., Patil, S. M., Kulkarni, M.D., Vijayaraghavan, N., Dongre, P. (2023). Efficient Net-XGBoost: An implementation for facial emotion recognition using transfer learning. Mathematics, 11(3): 776. https://doi.org/10.3390/math11030776

[17] Fard, A.P., Mahoor, M.H. (2022). Ad-Corre: Adaptive correlation-based loss for facial expression recognition in the wild. IEEE Access, 10: 26756-26768. https://doi.org/10.1109/ACCESS.2022.3156598

[18] Vulpe-Grigorasi, A., Grigore, O. (2021). Convolutional neural network hyperparameters optimization for facial emotion recognition. In 2021 12th International Symposium on Advanced Topics in Electrical Engineering (ATEE), Bucharest, Romania, pp. 1-5. https://doi.org/10.1109/ATEE52255.2021.9425073

[19] Reddi, P.S., Khrisna, A.S. (2023). CNN implementing transfer learning for facial emotion recognition. Intelligent Journal of Intelligent Systems and Applications in Engineering, 11(4S).

https://www.ijisae.org/index.php/IJISAE/article/view/2569.
[20] Huo, H., Yu, Y., Liu, Z. (2023). Facial expression recognition based on improved depthwise separable convolutional network. Multimedia Tools and Applications, 82(12): 18635-18652. https://doi.org/10.1007/s11042-022-14066-6
[21] Minaee, S., Abdolrashidi, A. (2019). Deep-emotion: facial expression recognition using attentional convolutional network. Sensors, 21(9): 3046. https://doi.org/10.3390/s21093046
[22] Goodfellow, I.J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Bengio, Y. (2014). Challenges in representation learning: A report on three machine learning contests. In International Conference on Neural Information Processing, Berlin, Heidelberg, pp. 117-124. https://doi.org/10.1007/978-3-642-42051-1_16
[23] Schroff, F., Kalenichenko, D., Philbin, J. (2015). FaceNet: A unified embedding for face recognition and clustering. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, pp. 815-823. https://doi.org/10.1109/CVPR.2015.7298682