# Convolutional Neural Network Model Comparisons For Face Emotion Recognition

Burcu Alakuş-Çınar
*Cognitive Science Department*
*Middle East Technical University*
Ankara, Türkiye
burcu.alakus@metu.edu.tr

***Abstract*—This study compares various Convolutional Neural Network (CNN) architectures and parameter configurations for face emotion recognition. Multiple CNN models were evaluated using a facial emotion dataset, adjusting key parameters. The results indicate significant variations in accuracies among the models. These findings offer insights into optimising CNNs for emotion recognition, with applications in human-computer interaction and security systems. The project codes can be found in the GitHub Repository [1].**

***Keywords*—convolutional neural network (CNN), face emotion recognition, emotion detection, deep learning, parameter tuning**

## I. Introduction

Face emotion recognition is a challenging and essential task in the fields of computer vision and artificial intelligence. It involves identifying and classifying human emotions from facial expressions captured in images or videos. Accurate emotion recognition has wide-ranging applications, including human-computer interaction, psychological analysis, security systems, and entertainment.

### A. Problem Statement

Despite significant advancements in deep learning, accurately recognising emotions from facial expressions remains a complex problem due to variations in facial features, lighting conditions, occlusions, and individual differences in expressing emotions. Traditional machine learning approaches often fail to handle these challenges, necessitating more sophisticated techniques such as CNNs.

### B. Objectives

This study, conducted as part of a deep learning course project, aimed to design, implement, and compare various CNN architectures and parameter configurations to identify the optimal model for face emotion recognition. By systematically adjusting key parameters like learning rate and model architectures, this research aims to develop deep learning models and understand their effects on facial emotion detection. The insights gained from this study contribute me a deeper understanding of how CNN models function in emotion recognition tasks."

## II. Related Work

### A. Studies on Face Emotion Recognition

Face emotion recognition has been an active area of research for several decades. Early approaches relied on handcrafted features and traditional machine-learning algorithms. However, these methods often struggled with variations in lighting, occlusions, and the subtlety of human emotions.

With the emergence of deep learning, CNNs have become state-of-the-art in face emotion recognition thanks to their ability to automatically learn hierarchical feature representations from raw images. Several notable studies have made significant contributions to this field:

Viola and Jones (2001) [3] introduced a powerful face detection framework that eases subsequent emotion recognition research. Their work focused on detecting faces in real time using Haar-like features and classifiers. Fasel and Luettin (2003) [4] provided a comprehensive review of early face emotion recognition methods, highlighting the limitations of traditional approaches and the potential of emerging techniques. Kahou et al. (2013) [5] demonstrated the effectiveness of CNNs for emotion recognition by combining deep features with temporal information from videos. Their model achieved high accuracy on benchmark datasets, displaying the potential of deep learning in this domain. Mollahosseini et al. (2016) [6] developed a deep neural network called 'AffectNet' that achieved state-of-the-art performance on the challenging FER-2013 dataset. Their work highlighted the importance of large-scale annotated datasets and data augmentation techniques for training deep models. Li et al. (2018) [7] proposed an attention mechanism to enhance the performance of CNNs in emotion recognition. Their model selectively focused on salient regions of the face, improving accuracy by reducing the effect of irrelevant background information. Wang et al. (2020) [8] investigated the impact of different CNN architectures and training strategies on emotion recognition performance. Their comprehensive study provided valuable insights into the trade-offs between model complexity and accuracy.

### B. CNN Architectures in Emotion Detection

CNNs have revolutionised the field of emotion detection from facial images resulting from their ability to automatically learn and extract features. Several CNN architectures have

been proposed and evaluated for their effectiveness in recognising facial emotions. Here is an overview of notable CNN architectures and their contributions to emotion detection:

AlexNet, is introduced by Krizhevsky et al.(2012) [9], significantly improved image classification performance. While not specifically designed for emotion recognition, its success in general image classification tasks helped many other models for its application in various domains, including emotion detection. Simonyan and Zisserman (2014) [10] proposed VGGNet, which emphasised the importance of deeper networks with smaller convolutional filters. VGGNet's architecture has been widely adopted in emotion recognition tasks due to its simplicity and effectiveness in capturing features. He et al. (2015) [11] introduced the ResNet architecture, which addressed the degradation problem in deep networks through residual learning. ResNet's ability to train very deep networks without performance loss has made it a popular choice for emotion recognition, providing high accuracy. Szegedy et al. (2015) [12] proposed the Inception architecture, which introduced the concept of multi-scale processing through Inception modules. This architecture has been effective in capturing diverse features at different scales, making it suitable for emotion detection tasks where facial expressions vary widely. Huang et al. (2017) [13] introduced DenseNet, which connects each layer to every other layer in a feed-forward fashion. This architecture favours feature reuse and eases the vanishing gradient problem, leading to improved performance in emotion recognition by leveraging dense connections. Minaee and Abdolrashidi (2017) [14] specifically designed the EmotionNet architecture for facial emotion recognition. This model combined CNNs with facial landmark localisation to enhance emotion detection accuracy. EmotionNet demonstrated the effectiveness of integrating spatial facial features with deep learning models. Howard et al. (2017) [15] developed MobileNet, a lightweight CNN architecture optimised for mobile and embedded applications. MobileNet's efficiency and low computational requirements have made it an attractive choice for real-time emotion recognition on resource-constrained devices. Li et al. (2022) [15] incorporated attention mechanisms into CNNs, allowing the network to focus on salient regions of the face. This approach significantly improved emotion recognition performance by emphasizing important facial features and reducing the influence of irrelevant background information.

## III. Methodology

### A. Dataset Description

For this study, a well-known facial emotion dataset, FER-2013 [2], that includes a diverse range of facial expressions labelled with corresponding emotions is used. The dataset consists of gray-scale images, each of size 48x48 pixels, categorised into seven emotion classes: anger, disgust, fear, happiness, sadness, surprise, and neutral. The dataset is split into training, validation, and test sets to facilitate model evaluation and comparison.


Fig. 1. Dataset Sample from FER-2013

### B. CNN Model Architectures

Several CNN architectures are designed and implemented, varying in depth and complexity, to evaluate their performance in facial emotion recognition. The architectures ranged from simple models with a few convolutional layers to more complex ones incorporating advanced techniques such as residual connections, separable convolutions, and transfer learning, also. The primary architectures used in this study include:

**Custom CNN with different learning rates:** A custom CNN model consists of several convolutional layers with ReLU activation and batch normalisation, followed by max-pooling and dropout layers, conclusive in fully connected dense layers, and outputs through a Softmax layer for classification as shown in Fig. 2 trained with learning rates of 1, 0.01, and 0.0001.

**Custom CNN using data augmentation:** Same architecture as custom CNN but trained with augmented data including re-scaling, random rotations, shifts, shearing, zooming, and horizontal flips, while also splitting 20% of the data for validation, with a learning rate of 0.0001.

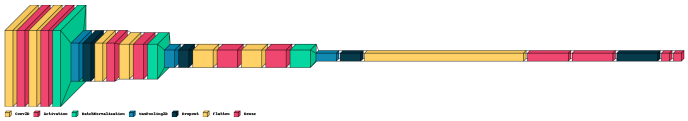

Fig. 2. Custom CNN architecture

**Simpler Custom CNN:** A simpler CNN model as in Fig. 3 includes three sets of convolutional layers with ReLU activation, batch normalisation, max-pooling, and dropout, followed by a flattened layer, a dense layer with 512 neurons, and a Softmax output layer.

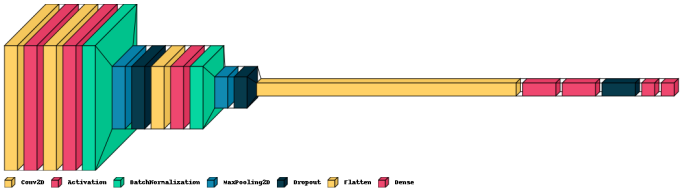


Fig. 3. Simpler custom CNN architecture

**Deeper Custom CNN:** A model includes multiple sets of convolutional layers with ReLU activation, batch normalisation, max-pooling, and dropout, with an additional set of convolutional layers for increased depth, followed by a flattened layer, a dense layer with 2048 neurons, and a Softmax output layer as seen in Fig. 4.

Fig. 4. Deeper custom CNN architecture

**Custom CNN using Residual Blocks:** This model features initial convolutional layers followed by multiple residual blocks with ReLU activation, batch normalisation, max-pooling, and dropout layers, ending in a flattened layer, a dense layer with 1024 neurons, and a Softmax output layer for classification, as seen in Fig. 5.

Fig. 5. Custom CNN using Residual Blocks

**Custom CNN using Separable Convolutions:** This CNN model utilises separable convolutions, as seen in Fig. 6 layers with ReLU activation, batch normalisation, max-pooling, and dropout layers, followed by a flattened layer, a dense layer with 1024 neurons, and a Softmax output layer for classification.

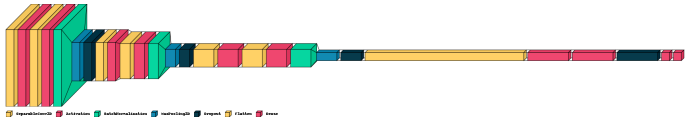

Fig. 6. Custom CNN using Separable Convolutions

**Custom CNN using Global Average Pooling:** This CNN model in Fig. 7 includes convolutional layers with ReLU activation, batch normalisation, max-pooling, and dropout, followed by a global average pooling layer, a dense layer with 1024 neurons, and a Softmax output layer for classification.

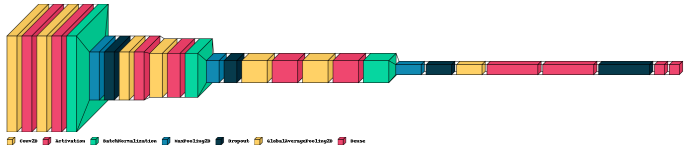

Fig. 7. Custom CNN using Global Average Pooling

**Transfer Learning with VGG16:** This is the model utilising VGG16 base (with pre-trained ImageNet weights and non-trainable layers) for feature extraction, followed by custom fully connected layers with ReLU activation and dropout for regularisation, ending in a Softmax output layer for classification as seen in Fig. 8

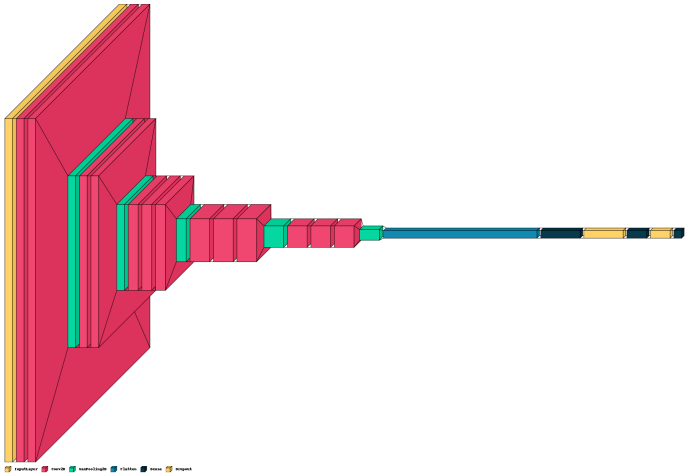

Fig. 8. Transfer Learning with VGG16

**Transfer Learning with ResNet50:** This model seen in Fig. 9 utilises the ResNet50V2 base (with pre-trained ImageNet weights and the last 50 layers trainable) for feature extraction, followed by custom layers including dropout, batch normalisation, a dense layer with ReLU activation, and a Softmax output layer for classification.

Fig. 9. Transfer Learning with ResNet50

### *C. Parameter Settings and Configurations*

In this study, the models were optimised using specific parameters and configurations. The images were 48x48 pixels and processed in gray-scale. The batch size was set to 64, and the number of epochs varied between 10, 50, and 100 to assess training duration impacts. Data pre-processing involved re-scaling pixel values to [0, 1], and the ImageDataGenerator was used for creating training and validation datasets with an 80-20 split. The Adam optimiser with a 0.0001 learning rate was employed, and categorical cross-entropy was used as the loss function. Key callbacks included ModelCheckpoint for saving the best model, EarlyStopping to prevent overfitting, ReduceLROnPlateau for dynamic learning rate adjustment, and CSVLogger for logging training data. The training process incorporated these elements to ensure robust model optimisation and performance evaluation.

### *D. Training and Validation Process*

Following the parameter setting and configurations, the models' performance was tracked through training and validation accuracy and loss metrics, which were plotted to visualise learning progress in Figures between 10-20. Final evaluations included test dataset performance, confusion matrices, and classification reports to provide detailed insights into model accuracy and classification capabilities in the codes.

### E. Hardware and Software Environment

The training and evaluation of the models were conducted on Google Colab Pro, which includes an NVIDIA Tesla T4 GPU. The Tesla T4 GPU enables efficient handling of large-scale deep learning tasks. The GPU was utilised to accelerate the training process, leveraging its high parallel processing capabilities to manage the computational operations required by the CNN models.

### F. Performance Metrics

For this study, training accuracy, validation accuracy, and loss values are considered for performance evaluations of the models. It's crucial to focus on validation accuracy in this case. High training accuracy with low validation accuracy might indicate over-fitting, where the model memorises the training data but struggles with unseen faces. Validation accuracy reflects the model's ability to recognise emotions in faces it hasn't encountered during training. This is the primary metric for evaluating the model's real-world performance. It is also important to look for models with lower losses, especially the validation loss. A lower loss indicates the model's predictions are closer to true emotions.

## IV. Results and Discussion

When first looking at Table I, the best model seems to be the transfer learning model with ResNet50, which has the highest validation accuracy and lowest loss. Transfer learning significantly enhances model performance. Image augmentation significantly enhances the model performance when one considers the custom CNN model and some models using the augmented data, with better training & validation accuracy while lowering the loss significantly.

Simplifying the model too much can lead to under-fitting, while deeper models require careful regularisation to avoid over-fitting. Lower learning rates (0.0001) generally yield better results, improving both training stability and accuracy.

### A. Model Performance Comparisons

From Figures 10, 11, and 12, where we can see the effect of different learning rates with the same model, one can see that the model with a learning rate of 0.0001 shows the most stable and consistent performance, with consistently increasing accuracy and decreasing loss. Higher learning rates (1 and 0.01) cause instability and divergence, as seen in the fluctuating accuracy and increasing loss. The model with a learning rate of 0.0001 performs the best in terms of both training and validation metrics, indicating it is the most suitable learning rate for this task. This comparison shows the importance of selecting an appropriate learning rate for training neural networks, as it significantly impacts model stability, convergence, and overall performance.

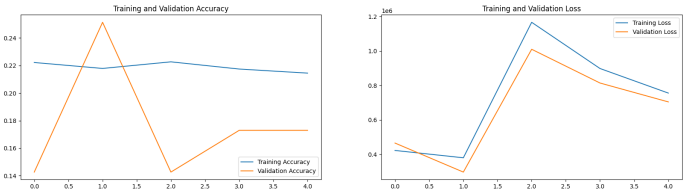


Fig. 10. Train/Validation Accuracy & Loss Graphs for Custom CNN Trained with Learning Rate of 1

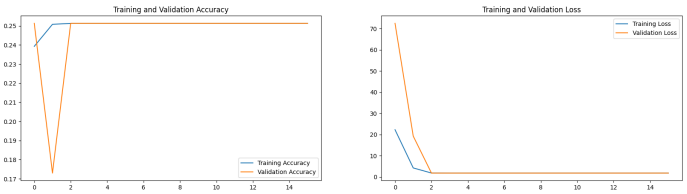


Fig. 11. Train/Validation Accuracy & Loss Graphs for Custom CNN Trained with Learning Rate of 0.01

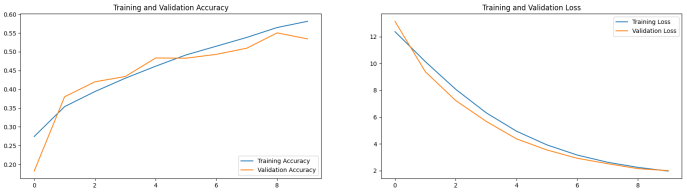


Fig. 12. Train/Validation Accuracy & Loss Graphs for Custom CNN Trained with Learning Rate of 0.0001

From Figures 12 and 13, the model trained with augmented data shows a more gradual and steady increase in accuracy, ultimately achieving higher overall accuracy. The validation accuracy in the augmented model is more stable and closely follows the training accuracy, suggesting better generalisation. The augmented model exhibits a consistent decrease in loss with lower final values, indicating more effective learning. The alignment of training and validation loss in the augmented model suggests less over-fitting compared to the non-augmented model. Data augmentation improves the model's ability to generalise to unseen data, as evidenced by the reduced gap between training and validation accuracy and loss.

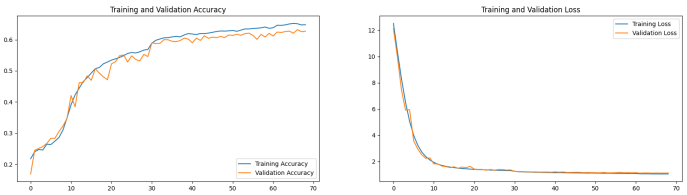


Fig. 13. Train/Validation Accuracy & Loss Graphs for Custom CNN Trained with Augmented Data

To compare the custom CNN, custom simpler CNN, and custom deeper CNN models, we need to check Figures 12, 14, and 15. For the task and dataset, the deeper model achieves

slightly higher training and validation accuracy compared to the simpler model, indicating that the additional layers allow it to capture more complex features and improve performance. The deeper model demonstrates a more significant decrease in loss, stabilizing at lower values, which indicates more effective optimisation and potentially better learning capability. Both models exhibit good generalisation, as evidenced by the close alignment of training and validation curves. However, the deeper model shows a slight edge in terms of lower loss and higher accuracy.

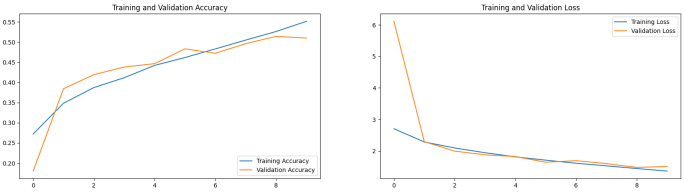


Fig. 14. Train/Validation Accuracy & Loss Graphs for Simpler Custom CNN

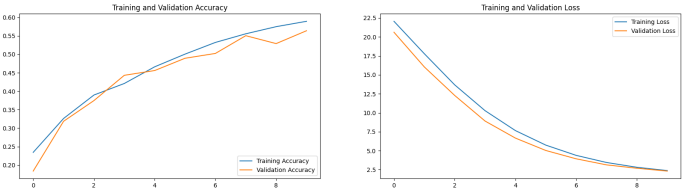


Fig. 15. Train/Validation Accuracy & Loss Graphs for Deeper Custom CNN

To compare models with residual blocks, separable convolutions, and global average pooling, we may first address their purposes for this task. Residual blocks are designed to minimise the vanishing gradient problem by allowing gradients to flow through the network more easily. They enable the training of much deeper networks by adding shortcut connections that bypass one or more layers. Separable convolutions aim to reduce the number of parameters in the model and computational cost while maintaining performance. They achieve this by decomposing standard convolutions into depth-wise and point-wise convolutions. Global average pooling replaces the fully connected layers at the end of the network with an averaging layer that reduces each feature map to a single value. This helps reduce over-fitting by decreasing the number of parameters and encourages the network to learn more general features.

By looking at Figures 16, 17, and 18, all three models show good generalisation, with validation accuracy closely following training accuracy. The model with separable convolutions achieves the highest accuracy, followed by the models with global average pooling and residual blocks. None of the models show significant overfitting, as indicated by the close alignment of training and validation loss curves. The model with separable convolutions performs the best overall, likely due to the reduction in parameters and computational efficiency. The model with global average pooling also performs well, demonstrating its effectiveness in reducing overfitting. The model with residual blocks performs adequately but does not reach the same accuracy levels as the other two models, likely due to the increased complexity.

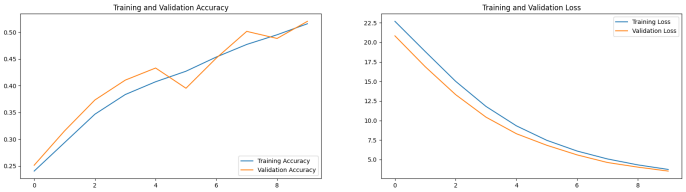


Fig. 16. Train/Validation Accuracy & Loss Graphs for Custom CNN with Residual Blocks

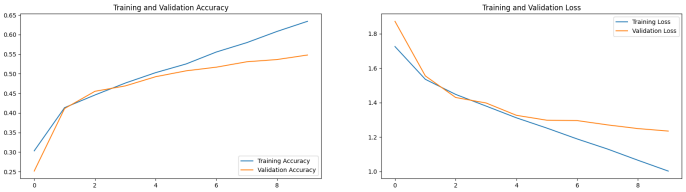


Fig. 17. Train/Validation Accuracy & Loss Graphs for Custom CNN with Separable Convolutions

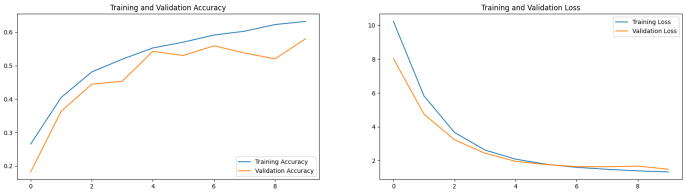


Fig. 18. Train/Validation Accuracy & Loss Graphs for Custom CNN with Global Average Pooling

For comparing transfer learning models, Figures 19 and 20 can be checked. The ResNet50 model achieves higher training and validation accuracy compared to the VGG16 model, suggesting that ResNet50 is better at capturing features relevant to the face emotion recognition task in this context. The VGG16 model demonstrates a more consistent and stable decrease in both training and validation loss, indicating effective learning and potentially better optimisation. Both models show good generalisation, with the validation accuracy closely following the training accuracy. However, the ResNet50 model shows slightly more fluctuations in validation accuracy and loss compared to VGG16, which might suggest a bit more sensitivity to over-fitting.

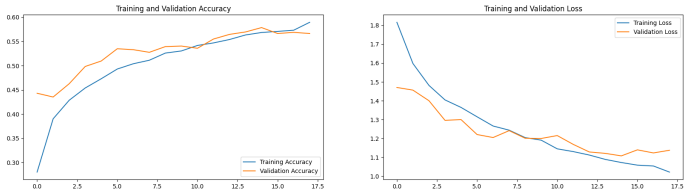


Fig. 19. Train/Validation Accuracy & Loss Graphs for Transfer Learning with VGG16

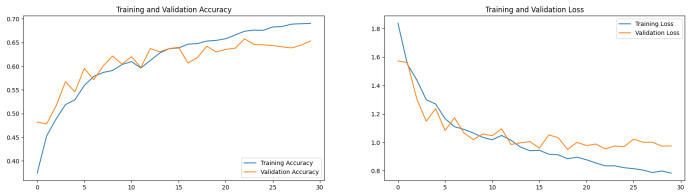


Fig. 20. Train/Validation Accuracy & Loss Graphs for Transfer Learning with ResNet50

### B. Analysis of Findings

From Table I, it can be seen that facial emotion recognition tasks benefit greatly from transfer learning. Pre-trained models like VGG16 and ResNet50 have already learned general image recognition features from massive datasets. Fine-tuning these models on a specific emotion recognition dataset allows them to leverage that knowledge and achieve better accuracy compared to training a CNN entirely from scratch with the same learning rate when we consider the validation accuracy and loss value. The image augmentation technique helps artificially expand the dataset by creating variations of existing images (flips, rotations, brightness adjustments). In face emotion recognition, it helps the model become more robust to variations in lighting, pose, and facial expressions not present in the original dataset. This can lead to improved validation accuracy when we compare it to the other validation accuracy in the table. For the impact of the model architectures that incorporate features specifically useful for facial recognition, the smaller filters in earlier layers seem to help capture facial features, which are often smaller and require finer details. Smaller filters in the initial convolutional layers can help with this. Dropout layers seem to help prevent over-fitting by randomly dropping out a certain percentage of neurons during training, forcing the model to learn more robust features.

## V. Conclusion

In this study, I implemented and evaluated several CNN models to understand the basics of deep learning and its application to face emotion recognition. Through systematic experimentation with different architectures, learning rates, and regularisation techniques, I explored the impact of these parameters on model performance. The results demonstrated the importance of selecting appropriate parameters to get more stable and accurate models. This hands-on approach provided me with valuable insights into model optimisation and the practical challenges of training deep learning models, laying a strong foundation for further exploration.

### A. Future Work

For future work, other advanced architectures could be explored such as EfficientNet, DenseNet, or transformer-based models, to further improve the performance of facial emotion recognition systems. Investigating the impacts of fine-tuning different layers and employing transfer learning from models pre-trained can be considered. Also, experimenting with more sophisticated data augmentation techniques, such as Generative Adversarial Networks (GANs) for generating synthetic training data could be adapted to address class imbalances and improve model robustness.

The models can be extended to incorporate multimodal inputs, such as audio and physiological signals, to enhance the accuracy and reliability of emotion recognition systems with video datasets. One can also evaluate the models across multiple datasets to assess their generalisability and robustness to different types of facial expressions.

## References

[1] B. Alakuş, "di504-project-face-emotion-recognition," GitHub, 2024. [Online]. Available: https://github.com/burcia1711/di504-project-face-emotion-recognition

[2] M. Sambare, "Facial Expression Recognition (FER-2013) Dataset," Kaggle, 2018. [Online]. Available: https://www.kaggle.com/datasets/msambare/fer2013/data. [Accessed: 12-Jun-2024].

[3] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2001, vol. 1, pp. I-511–I-518.

[4] B. Fasel and J. Luettin, "Automatic facial expression analysis: A survey," Pattern Recognit., vol. 36, no. 1, pp. 259–275, Jan. 2003.

[5] S. R. J. Kahou, C. Pal, X. Bouthillier, P. Froumenty, Ç. Gülçehre, R. Memisevic, P. Vincent, A. Courville, and Y. Bengio, "Combining modality specific deep neural networks for emotion recognition in video," in Proc. ACM Int. Conf. Multimodal Interact., 2013, pp. 543–550.

[6] A. Mollahosseini, B. Hasani, and M. H. Mahoor, "AffectNet: A database for facial expression, valence, and arousal computing in the wild," IEEE Trans. Affect. Comput., vol. 10, no. 1, pp. 18–31, Jan.-Mar. 2019.

[7] S. Li, W. Deng, and J. Du, "Reliable crowdsourcing and deep locality-preserving learning for expression recognition in the wild," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2017, pp. 2852–2861.

[8] S. Wang, Y. Guo, Y. Yang, L. Yu, and C. Zhang, "Emotion recognition by a fully end-to-end deep neural network with attention mechanism," in Proc. Int. Joint Conf. Neural Netw., 2020, pp. 1–7.

[9] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in Proc. Advances Neural Inf. Process. Syst., 2012, pp. 1097–1105.

[10] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in Proc. Int. Conf. Learn. Represent., 2015.

[11] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2016, pp. 770–778.

[12] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2015, pp. 1–9.

[13] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2017, pp. 4700–4708.

[14] S. Minaee and A. Abdolrashidi, "Deep-emotion: Facial expression recognition using attentional convolutional network," arXiv preprint arXiv:1902.01019, 2019.

[15] A. G. Howard et al., "Mobilenets: Efficient convolutional neural networks for mobile vision applications," arXiv preprint arXiv:1704.04861, 2017.

[16] S. Li, W. Deng, "Deep facial expression recognition: A survey," IEEE Trans. Affect. Comput., vol. 13, no. 1, pp. 119-134, Jan.-Mar. 2022.

TABLE I
PERFORMANCE OF DIFFERENT CNN ARCHITECTURES

| Model | Learning rate | Train Accuracy (%) | Validation Accuracy (%) | Loss |
|---|---|---|---|---|
| **CNN from scratch** | 1 | 25.13 | 24.71 | 296462.0625 |
| | 0.01 | 25.13 | 24.71 | 1.8133 |
| | 0.0001 | 60.92 | 54.18 | 1.9963 |
| **CNN from scratch + Image Augmentation** | 0.0001 | 67.18 | 64.92 | 1.0603 |
| **Simpler CNN from scratch** | 0.0001 | 63.63 | 51.32 | 1.4897 |
| **Deeper CNN from scratch** | 0.0001 | 61.15 | 57.01 | 2.2941 |
| **CNN with Residual Blocks** | 0.0001 | 54.61 | 51.53 | 3.5216 |
| **CNN with Separable Convolutions** | 0.0001 | 71.80 | 55.54 | 1.2115 |
| **CNN with Global Average Pooling** | 0.0001 | 64.59 | 58.74 | 1.4685 |
| **Transfer Learning VVG16** | 0.0001 | 59.23 | 57.86 | 1.1077 |
| **Transfer Learning Resnet50** | 0.0001 | 70.11 | 65.78 | 0.9542 |