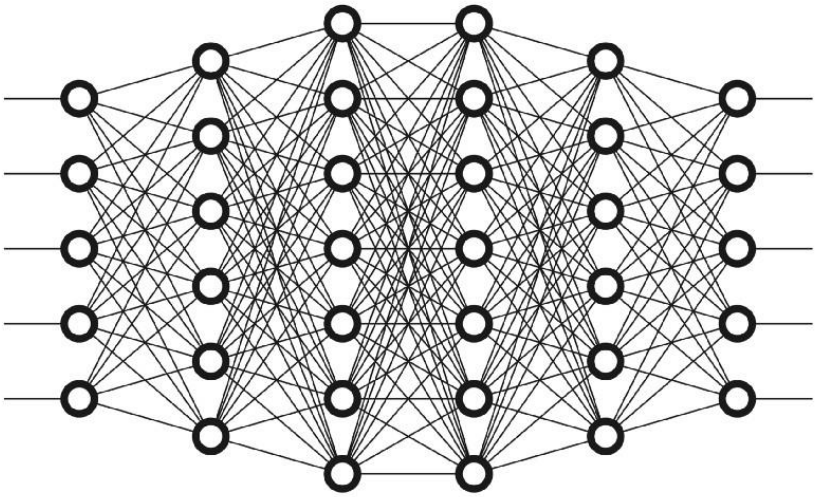
Final Project (CMPSC-448) Report

By: Mohammad Sabih [section-001] (mvs7196@psu.edu)
Abdullah Nadeem [section-001] (azn5413@psu.edu)



The entire implementation is done using the Keras+Tensorflow framework on Google Colab!

I. Introduction

In this project we chose to focus on Convolutional Neural Networks (CNNs) and Vision Transformers (ViTs), applying them to the task of facial emotion recognition. This decision was motivated by the distinct architectural differences between CNNs and ViTs, and our interest in evaluating their respective performances in accurately classifying facial expressions into various emotional states. The project aims to provide insights into how these two prominent deep learning architectures handle the complexities and nuances of emotion recognition from facial images.

II. Data

We have used the FER2013 dataset, which is specifically designed for facial emotion recognition and is publicly available. This dataset contains 28,709 grayscale images, each measuring 48x48 pixels. These images are categorized into seven different emotions: happiness, sadness, anger, fear, surprise, disgust, and neutrality, with happiness being the most predominant emotion represented. Fig.1 shows the sample images of the dataset. The uniqueness of FER2013 lies in its diversity. The images were sourced from Google image searches, encompassing a wide range of facial expressions. This variety includes different angles, lighting conditions, and emotional intensities, making the dataset quite challenging for emotion recognition models. FER2013 gained popularity after being featured in a Kaggle competition. It's a tough dataset for deep learning models, as evidenced by the fact that reported accuracies on this dataset usually range from 31% to 65%. This variation in model performance highlights the complexity of the dataset and the difficulties encountered in accurately recognizing and classifying facial emotions.

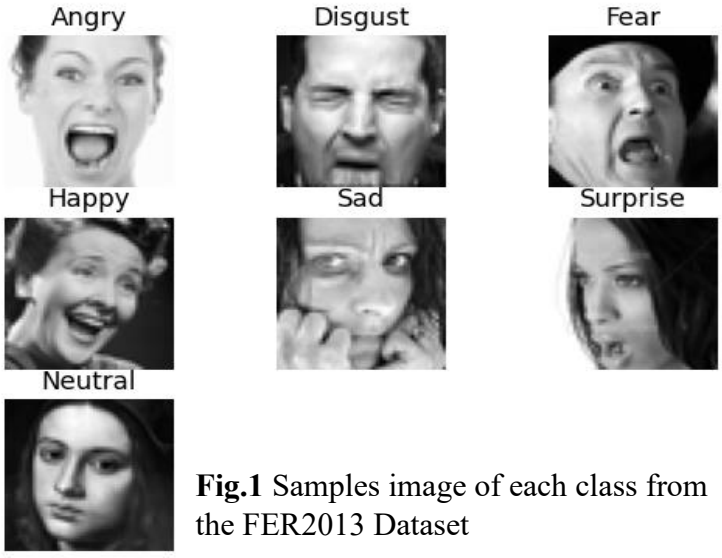


Fig.1 Samples image of each class from the FER2013 Dataset

2.1 Data Preprocessing

We first normalized the images in our dataset to facilitate more efficient model training. This normalization involved scaling the pixel values of each image by dividing them by 255.0, thereby converting them to a range between 0 and 1. This step is crucial as it helps in reducing the computational complexity and allows the models to converge faster during training. Furthermore, for the second stage of training, as will be discussed in section.5.3, we implemented data augmentation as a pre-processing step to enhance the diversity of our training set and to help fight the challenge of over-fitting.

- *Rotation*: Set to 20 degrees, allowing images to be rotated within this range. This introduces variability in the dataset by simulating different orientations of faces.
- *Width shift range* and *height shift range*: Both set to 0.1, enabling horizontal and vertical shifts of the images by 10% of their width and height, respectively. This simulates slight changes in the position of the face in the images.
- *Zoom range*: Set to 0.1, allowing a zoom of up to 10% on the images. This helps the model to learn from various scales of facial features.
- *Horizontal flip*: Set to True, allowing images to be flipped horizontally. This mimics the scenario of seeing a mirrored face, increasing the diversity of the dataset.

Fig.2 Shows the visual change in one of the dataset images, after processing through the addressed pre-processing steps.

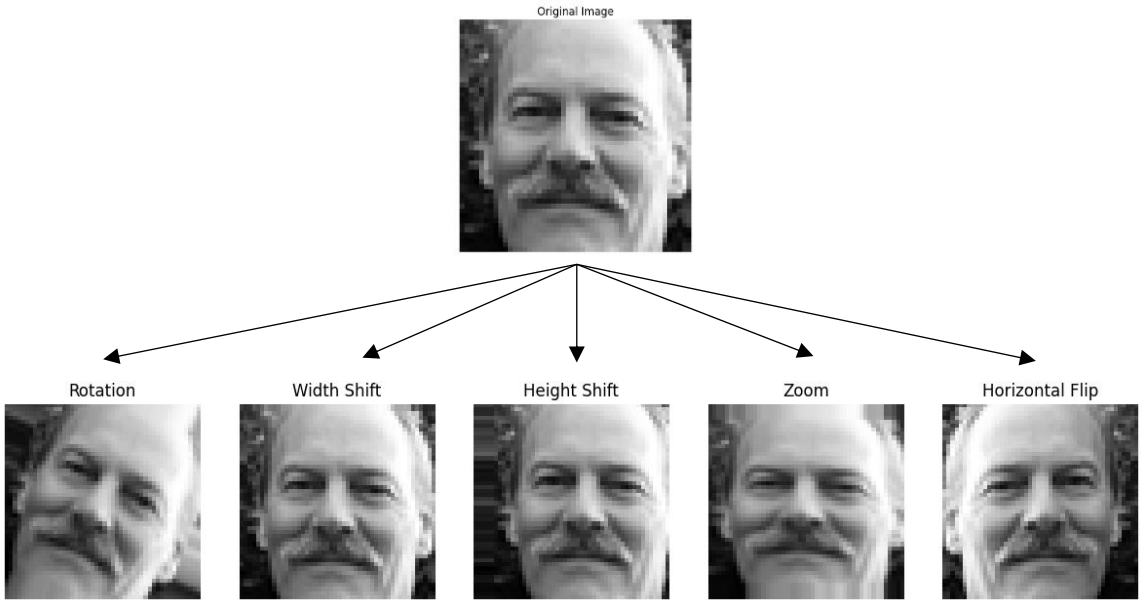


Fig.2 Visualization of pre-processed Transformation of an Image from the Dataset.

2.2 Data splitting

We aimed at maintaining an equal proportion of class samples in the training, validation, and test sets. To achieve this, we employed stratified splitting, a technique that ensures each set contains a representative distribution of the different emotion categories. Initially, we allocated 20% of the data to the test set, mirroring the overall emotional distribution of the complete dataset. We then divided the remaining data, using the same approach, into a final training set and a validation set, with 20% going to the validation set. This careful stratification resulted in balanced and representative datasets: 18,373 images for training, 4,594 for validation, and 5,742 for testing, providing a robust foundation for both training and evaluating our models.

III. Model Architecture

3.1 CNN (Convolutional Neural Network)

Our CNN model for emotion classification comprises of four convolutional layers, increasing in complexity from 64 to 512 filters. Each convolutional layer is followed by batch normalization and MaxPooling, using 3x3 kernels and 'relu' activation. To prevent overfitting, dropout layers with a rate of 0.3 are added. Post convolutional layers, the model features three fully connected dense layers with 512, 256, and 64 neurons, each also followed by dropout. The final output layer, with 7 neurons and a 'softmax' activation, corresponds to the seven emotion classes, providing the final classification. This design effectively captures and classifies facial expression features. Fig.3 shows the architecture of the CNN which we employed for learning our dataset.

3.2 ViT (Vision Transformer)

Our Vision Transformer (ViT) model for emotion classification adopts the traditional tranformer architecture. The key components of this ViT model include:

- Patch Embedding: This layer converts the input images into a sequence of flattened patches and projects them into a specified dimension (projection dim). It ensures each image patch contributes to the model's understanding of the overall image.
- Transformer Block: This is the core of the ViT, comprising multi-headed self-attention mechanisms and multi-layer perceptron's (MLP). The attention mechanism helps the model focus on different parts of the image, while the MLP processes these features. Layer normalization is applied for stability and efficiency.
- Global Average Pooling and Output: After passing through the specified number of transformer layers, the model applies global average pooling to the sequence of patch embeddings. This step is followed by a dropout layer for regularization and a dense output layer with a softmax activation function to classify the images into one of the seven emotion categories.

The ViT model we developed uses an input shape comprising of a single-color channel. It processes the images in patches of 4x4 pixels, resulting in 144 patches per image. We set the projection dimension to 16, with 4 attention heads and 4 transformer layers. This architecture is particularly adept at capturing and integrating global information from the entire image, leading to effective emotion classification. The hyperparameter tuning can be understood by section. 5.2. Fig.4 shows the architecture of the ViT which we employed for learning our dataset.

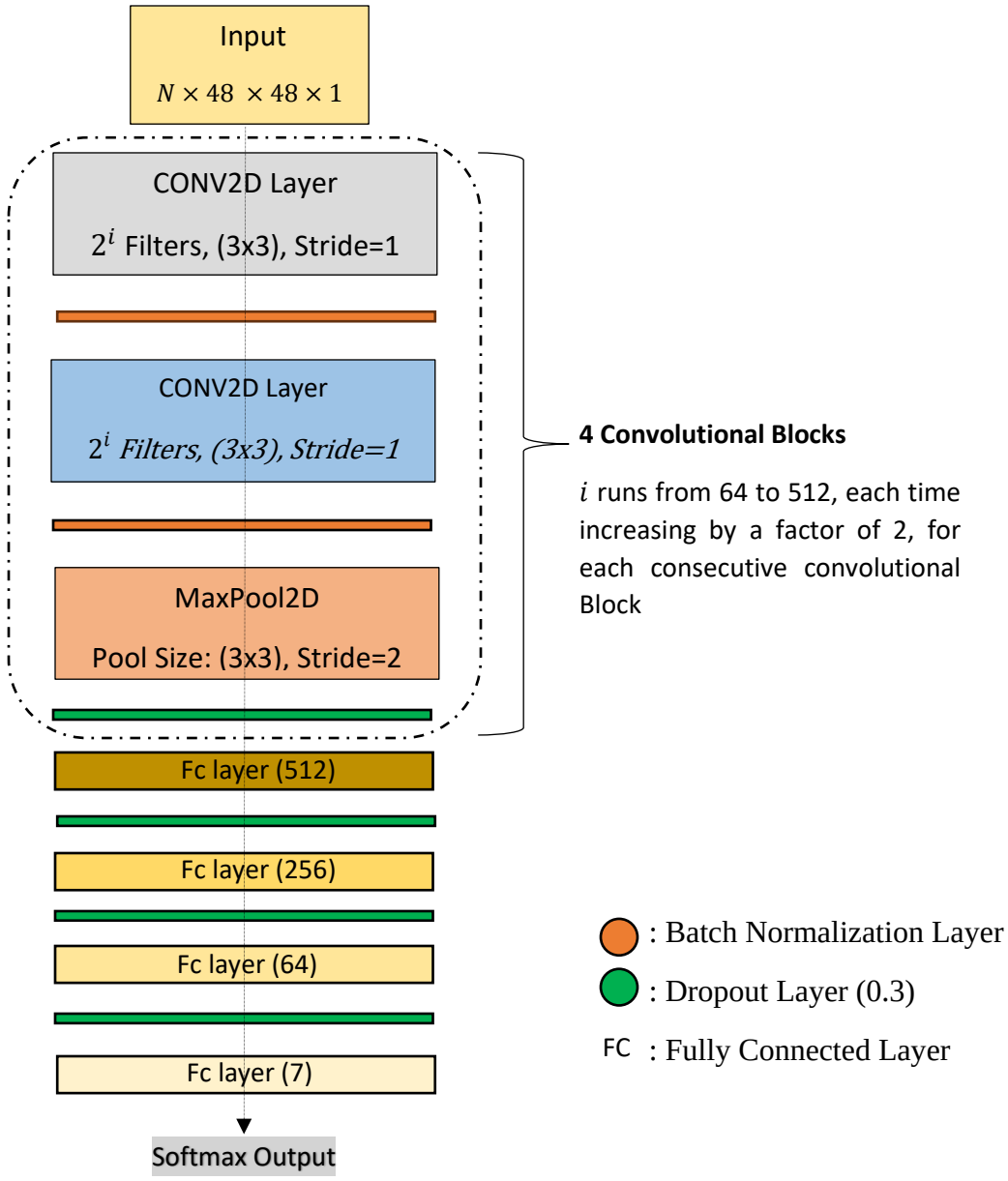


Fig.3 Convolutional Neutral Network Architecture.

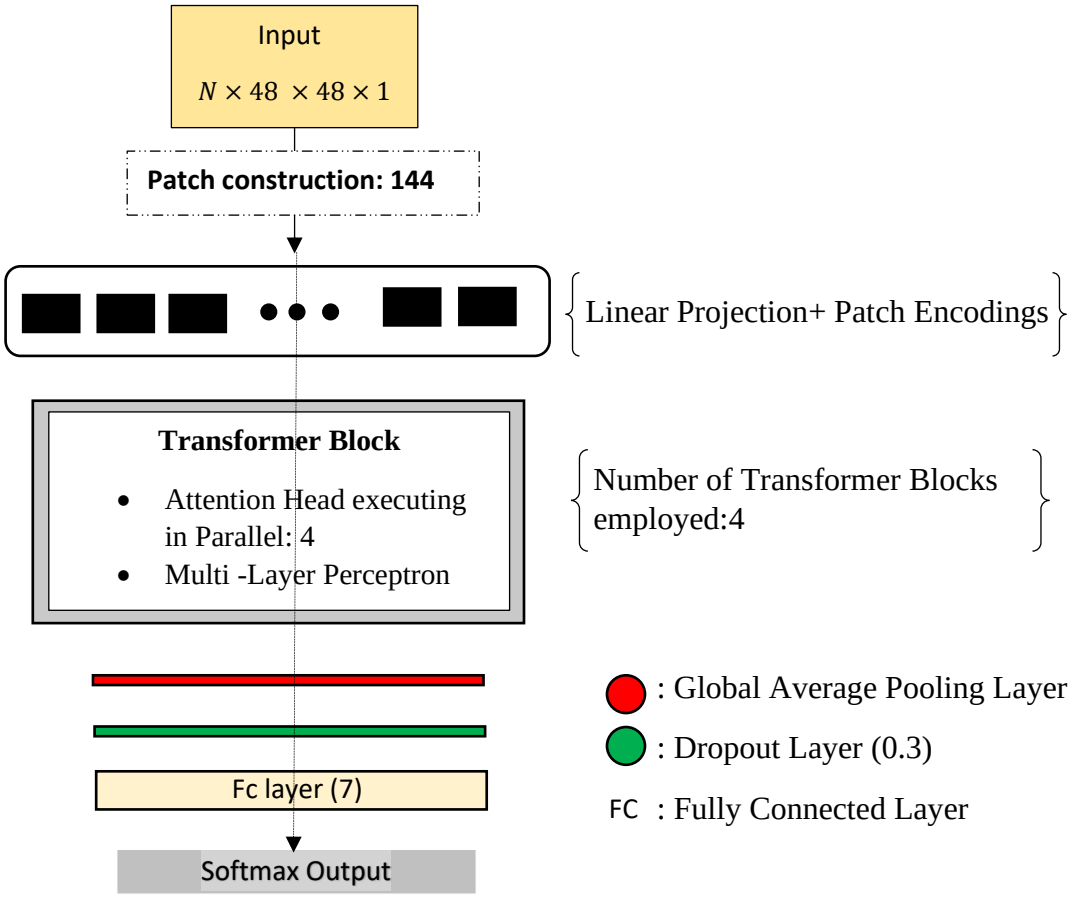


Fig.4 Vision Transformer for Image Classification Architecture

IV. Training Details

For training the complete network is trained end-to-end, i.e. the weights of the CNN as well as the VIT network are optimized using backpropagation using simple gradient descent. For loss optimization, the ADAM optimizer is used, and training sample parameters are updated as per Eq. (1)

$$\theta = \theta - \eta . \nabla_\theta J(\theta; x_i; y_i) \quad (1)$$

Here, η (initialized with 0.001) represents the learning rate and J the loss function which is, "$categorical\ cross\ entropy$" whose computational Eq. (2) is shown for batch size=n. (for both the models n is set to 64 while training)

$$Loss = -\sum_{i=1}^{n} y_i . \log(\hat{y}_i) \quad (2)$$

Where $\hat{y}_i$ and y_i are the predicted scalar and probable class value of an input x_i respectively.

In our training procedure, we implemented an Early Stopping mechanism to prevent overfitting. Overfitting is a common issue in deep learning where a model learns to replicate the training data too closely, losing its ability to generalize to unseen data. To mitigate this, we monitored the validation accuracy after each epoch. If the validation accuracy did not improve for a consecutive span of 10 epochs, the training process was automatically halted. Early Stopping thus serves as a safety mechanism against overfitting.

Another approach which we employed during training was smart learning rate reduction on plateaus. The learning rate is a crucial hyperparameter in the training of neural networks, determining the size of the steps taken during the optimization process. Too large a learning rate can cause the model to converge too quickly to a suboptimal solution, while too small a rate can lead to a prolonged training process. To strike a balance, we utilized a dynamic adjustment approach (Eq. (3)) where the learning rate was reduced by a factor of 0.9 if the validation loss did not improve for 5 consecutive epochs. By adapting the learning rate in response to the plateauing of validation loss, we aimed to achieve more precise convergence, thereby enhancing the model's performance.

$$\eta_{reduced} = \eta_{old} \times 0.9 \tag{3}$$

V. Experiments and Results

5.1 Training Statistics of CNN

During our training progression for the CNN model, we initiated with a learning rate of 0.001, maintaining this rate constant until the 24th epoch. The learning rate was progressively reduced by the *Keras callback* as we had configured it to automatically adjust in response to the model's performance metrics; therefore, the learning rate dropped down to 0.0009 at the 25th and stayed consistent till the 30th epoch, followed by a further reduction to 0.00081 for the next five epochs. These strategic adjustments were aimed at refining the model's learning as the training advanced. By the 42nd epoch, we observed significant progress: the training loss had diminished to 0.1858, and the model achieved a commendable training accuracy of 98.32%. However, the model's performance on the validation set revealed a loss of 1.9669 and an accuracy of 62.02%. This contrast between training and validation outcomes highlighted a challenge in the model's ability to generalize to new, unseen data, suggesting a potential overfitting issue. We have addressed the solution of this challenge in section5.3. Fig.5 shows the training curve.

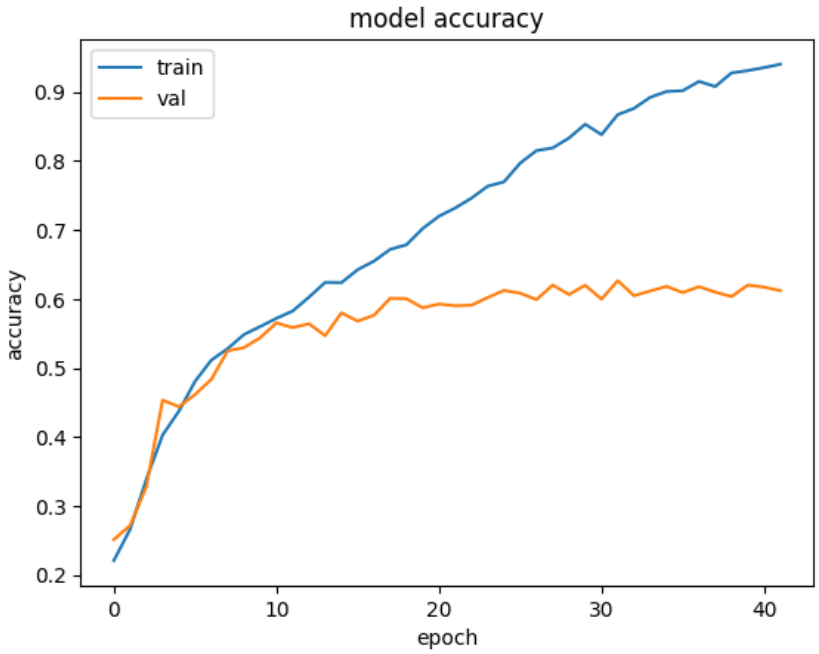


Fig.5 Training Curve of CNN

5.2 Training Statistics of VIT

During the training of our Vision Transformer (ViT) for image classification, we faced several challenges and had to engage in extensive hyper-parameter tuning. Initially, we configured the model with 6 transformer layers, 6 heads, and a projection vector size of 128. However, this setup did not yield significant learning, as indicated by the stagnating training accuracy.

In an attempt to improve performance, we first reduced the projection vector size to 64. This adjustment led to a slight improvement, but the results were still not up to our expectations. We then modified the architecture to 4 transformer layers and 4 heads while keeping the projection vector size at 64, but this change did not produce any noticeable difference.

Our breakthrough came when we further reduced the projection vector size to 32. This adjustment resulted in a significant improvement in model accuracy, suggesting that the model was effectively learning. Encouraged by this, we experimented with reducing the projection vector size to 16, which also yielded similar performance. Opting for computational efficiency, we decided to continue with a projection vector size of 16.

By the 51st epoch, the model demonstrated a training loss of 1.2982 and an accuracy of 52.02%, while the validation loss and accuracy were 1.6457 and 38.07% respectively. Throughout the training, the learning rate was progressively reduced from the initial 0.001 to 0.00053144, adapting to the model's learning needs and enhancing its performance. Fig.6 shows the training curve.

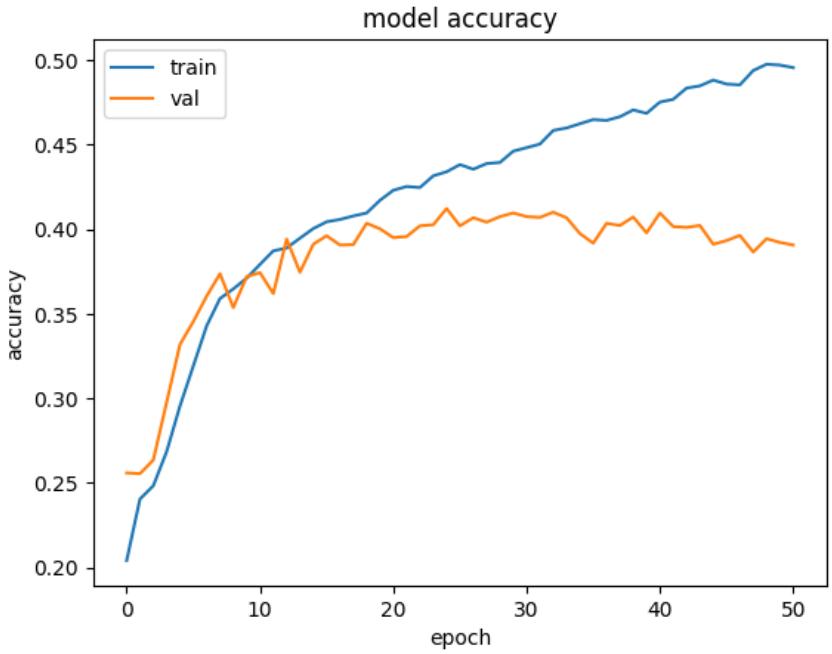


Fig.6 Training Curve of ViT

5.3 Solving the obstacle

After recognizing the issue of overfitting, as evidenced by the disparity in performance between training and validation data, we took a targeted approach to refine our model. We retrained it using the samples that were initially misclassified (*the hard samples*), giving the model exposure to these challenging instances. Therefore, the models CNN and ViT both were trained to a count of 20 epochs on their respective wrong hard sets from the first round of training. The changes observed over the validation set performance can be understood by Table (), which definitely shows how both of our models were capable of tweaking their weights to have a partial fit over the hard samples.

However, aware that this could potentially lead to overfitting on these specific samples, we planned a subsequent round of training along with data augmentation to achieve a balanced performance. This strategy aimed to broaden the diversity of our training dataset, helping the model to not only overcome the overfitting challenge but also to learn more generalized features from each image. Table.1 shows the results observed after this second step of training and Fig.7 shows the training curves, and it can be observed that for CNN, the training accuracy decreased to 80.42% with a training loss of 0.9100 which happened over 13 epochs. However, the validation accuracy improved to 66.52%, with the validation loss slightly lower at 0.8989. Similarly, the Vision Transformer (ViT) after being trained for 20 epochs resulted in a loss of 1.6035 and an accuracy of 42.95% and 41.51 on the train and validation set respectively.

It can be stated that for both the models change in performance, where the training accuracy decreased and attained a potentially good value when compared with the validation stats with the

simultaneous increase in the validation accuracy, suggests that both the models have become more generalized and less prone to overfitting.

Conclusively speaking since the models have become more generic now, they are bound to perform better on unseen data as can be seen from Table (1).

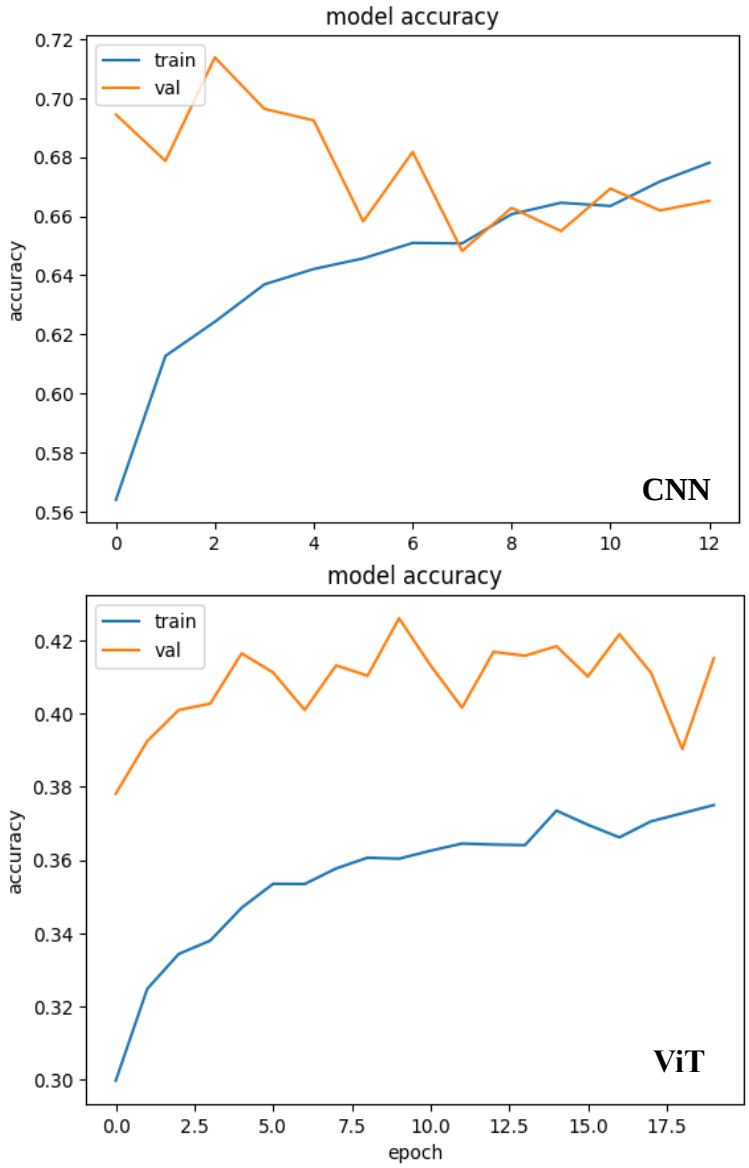


Fig.7 Training Curve of Training Phase-2 for both the Models.

Table.1 All Recorded Statistics before and after Encountering the over-fitting problem

	Models	
Statistics	**CNN**	**ViT**
Training Accuracy after Phase-1	0.9832	0.5202
Validation Accuracy after Phase-1	0.6202	0.3807
Test Accuracy after Phase-1	06289	0.3962
Training Accuracy after Phase-2	0.8042	0.4295
Validation Accuracy after Phase-2	0.6652	0.4151
Test Accuracy after Phase-2	0.6472	0.4187
Reported final Test Set Loss	0.9814	1.5021

VI. Comparison

6.1 Statistical Comparison

6.1.1 Training Phase-1 (Before Data Augmentation and Hard Samples Training)

- *Training Accuracy:* The CNN significantly outperformed the ViT with an accuracy of 98.32% compared to ViT's 52.02%. This suggests the CNN was more effective in learning from the training data.
- *Validation Accuracy:* The CNN also had higher validation accuracy (62.02%) than the ViT (38.07%), indicating better generalization capabilities at this stage.
- *Test Accuracy:* Similar trends were observed in test accuracy, with the CNN achieving 62.89% and ViT 39.62%.

6.1.2 Training Phase-2 (After Data Augmentation and Hard Samples Training)

- *Training Accuracy:* Both models saw a decrease in training accuracy, more so for the CNN (from 98.32% to 80.42%) than for the ViT (from 52.02% to 42.95%). This suggests that the models were trained on more challenging data, reducing overfitting.
- *Validation Accuracy:* Both models improved in validation accuracy, with CNN reaching 66.52% and ViT 41.51%, indicating enhanced generalization to unseen data.
- *Test Accuracy:* Similar to validation accuracy, both models showed improvement in test accuracy, with CNN at 64.72% and ViT at 41.87%.
- *Decrease in Test Set Loss*: The CNN showed a significant decrease in test set loss (0.9814) compared to the ViT (1.5021), indicating a more substantial reduction in the error rate of the CNN.

Speaking conclusively, CNN consistently outperformed the ViT across all metrics in both training phases. It demonstrated higher accuracy and a more substantial decrease in loss, indicating its better suitability for this particular emotion classification task. The improvements in validation and test accuracies in the second phase for both models suggest that data augmentation and training

on hard samples effectively reduced overfitting, leading to models that generalize better to new data.

6.2 Class-Wise Comparison

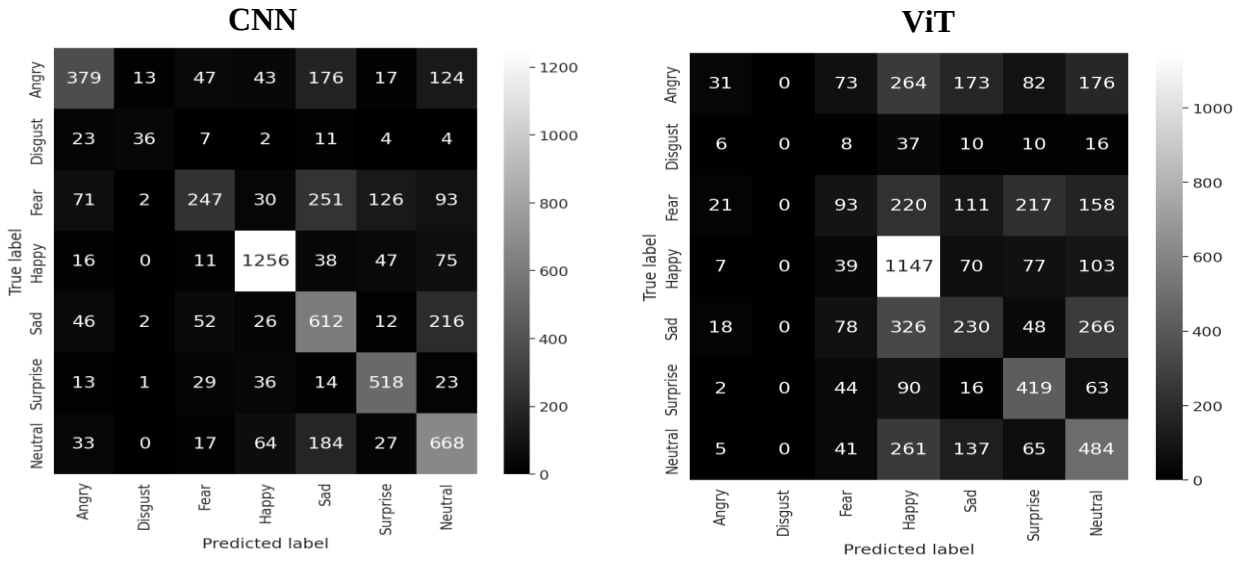


Fig.8 The confusion matrix of both CNN, ViT models obtained via Test Set.

The confusion matrix obtained for both models over the Test-set can be realized with Fig.8 and it can be interpreted that both models performed exceptionally well in predicting the 'Happy' class, which is understandable since it constitutes approximately 24% of the dataset. The 'Angry' class was better learned by the CNN, achieving a total of 379 correct predictions, compared to the ViT, which only managed 31 correct predictions. For the 'Surprise' and 'Neutral' classes, the performance of both models was nearly identical. However, for the 'Disgust' class, which was the toughest to learn due to its exceptionally low number of samples compared to the other six classes, the CNN managed to correctly predict 36 instances, while the ViT could not make a single correct prediction and often misclassified 'Disgust' samples as 'Happy'.

6.3 Feature Comparison

We found that both models properly justify the feature learning expected trends. First speaking for CNN , We know the first few layers of a CNN typically capture basic image features such as edges, colors, and textures. These layers act as basic image filters and often retain much of the spatial and structural information present in the original image. Therefore, we should be able to recognize the overall shape or parts of the original image from the feature maps of earlier layers, also we know as we move deeper into the network, the layers start to represent more abstract features. These

layers combine the basic features detected by earlier layers to identify more complex patterns, such as parts of objects or even entire objects. However, these feature maps are less visually interpretable in terms of the original image content. Therefore, they should tend to look more like 'random stuff' because they represent high-level features that are not visually recognizable to humans. Fig.9 shows the visual representation of the feature maps learned by our earlier and later layer of CNN and it aligns with the expected trend.

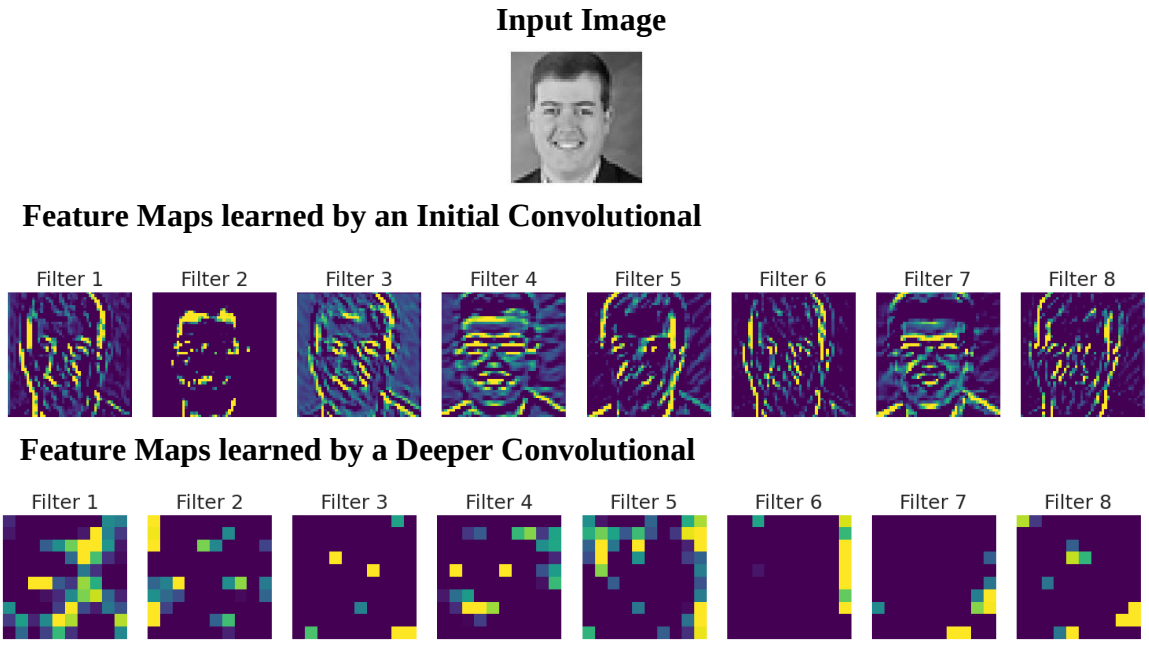


Fig.9 Visual Representation of the feature maps spread over 5 filters as are obtained from the Trained CNN after processing a sample Image.

Now talking about ViT, we know the earlier transformer layers usually process more basic patterns and features extracted from the input image. Therefore the activations might reflect simpler or more general aspects of the image data, as these layers are closer to the raw input. Similarly since deeper layers begin to combine the basic features into more complex and abstract representations. The activations of these layers should reflect higher-level features that are more specific to the particular task. Therefore if we plot the mean activation across patches over the trained ViT, the bar plot of the first layer should potentially show relatively uniform activations, as the layer is still processing basic aspects of the image. In contrast, the bar plot of the fourth layer should show more varied activations, reflecting the complexity and specificity of the features it is learning. Our Fig. 10 completely aligns with this understanding.

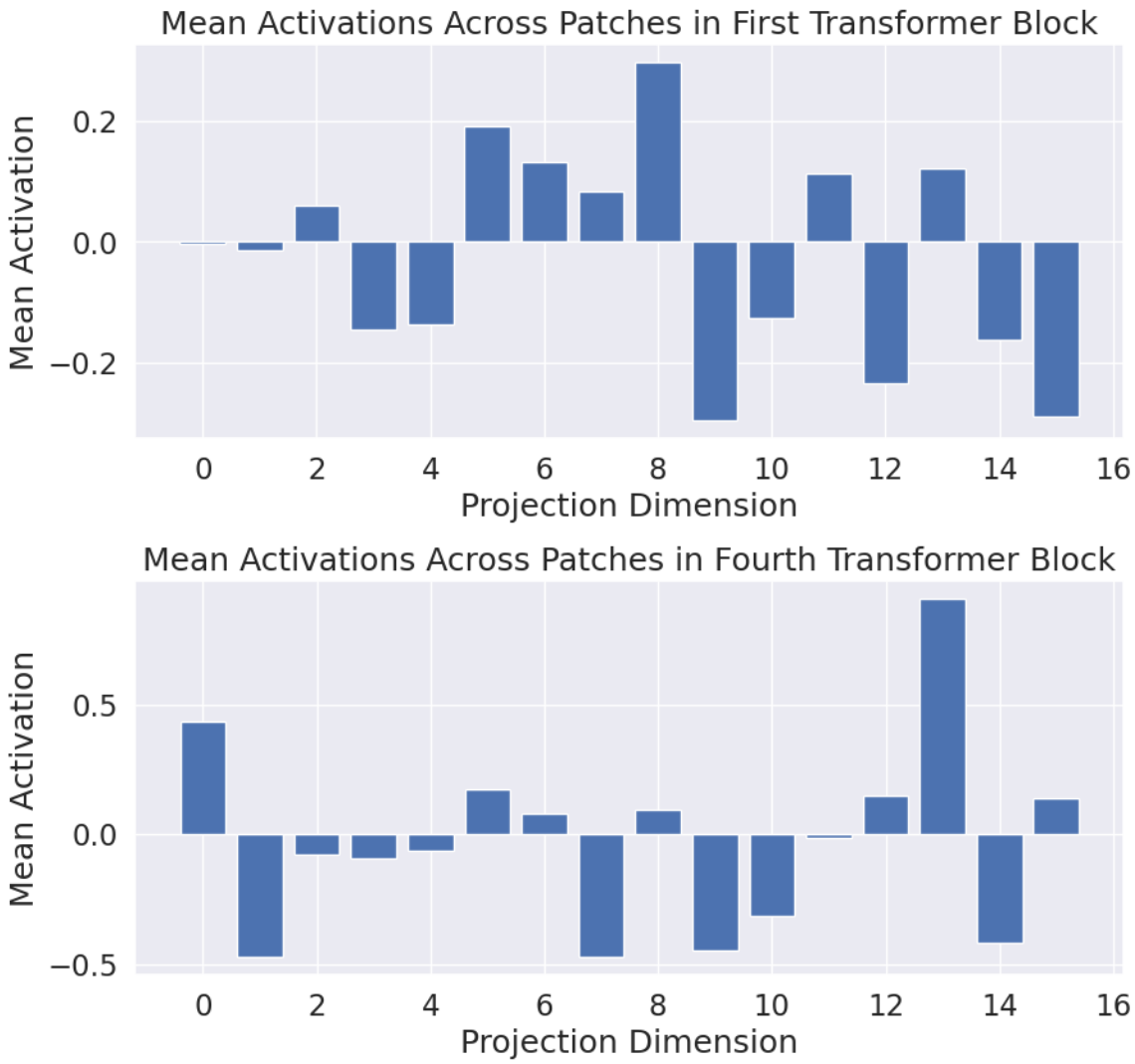


Fig.10 Bar plot of the Mean activation distribution as is reported by the Trained ViT model over the 16 length projection vector when processed through the first and fourth Transformer Block

VII. Theoretical Interpretation and Conclusion

In our facial emotion recognition task, which involves relatively small images (48x48), the Convolutional Neural Network (CNN) outperformed the Vision Transformer (ViT) for several interrelated reasons. Firstly, CNNs have built-in features that are really good for understanding images. They can handle shifts and changes in pictures and focus on small, detailed parts, which helps them learn from smaller images effectively. This inherent design makes CNNs more data-efficient, a crucial advantage in scenarios where each image contains limited information. Additionally, CNNs are particularly skilled at extracting hierarchical features from images. This ability is especially beneficial for small-sized images where understanding and capturing local details are paramount. In contrast, ViTs, with their self-attention mechanisms, generally brings in higher computational overhead. While self-attention is advantageous for larger images with

complex patterns, its benefits are less pronounced in small images. Moreover, ViTs process images in patches, a technique that can lose its effectiveness in very small images due to insufficient patch granularity to capture meaningful information. These factors collectively explain why for the task we dealt with CNNs out-performed ViT.

VIII. Future Work

- Since we found data imbalance for the class ‘Disgust’, in the future experiments we aim to tweak the loss function for a potentially smooth training. Simply implying, increasing the penalty of imposed loss on the wrong prediction of the Class ‘Disgust’ should be high so model learns it more carefully. We believe this strategy can further help in removing the reported accuracy.
- Since some visual features are highly interrelated, and can be really random for any random human, therefore adopting an unsupervised method to solve the problem will be more efficient. Therefore we aim to extract feature representations from the CNN and then put into use one of the most fit and sound clustering algorithm to target the classification.