Experiments on Facial Emotions Recognition using Deep Learning

1st Roisul Islam Rumi
School of Computer Science
The University of Windsor
Ontario, Windsor
rumir@uwindsor.ca

2nd Lian Duan
School of Computer Science
The University of Windsor
Ontario, Windsor
duan12@uwindsor.ca

3rd Naga Jyothirmayee Dodda.
School of Computer Science
The University of Windsor
Ontario, Windsor
doddan@uwindsor.ca

4th Sumaiya Deen Muhammad.
School of Computer Science
The University of Windsor
Ontario, Windsor
deenmuh @uwindsor.

a. Abstract

Human emotion recognition is one of the most sought-after research topics in computer vision, particularly in the facial feature extraction space. To tackle this problem Convolutional Neural Networks (CNN) has been a well-known approach in the field of Computer Vision. Image classification is one of the most important and high-demand areas of Computer Vision systems. Numerous image classification models have been developed to improve recognition accuracy. In this paper, we have proposed two CNN architectures, and Vision Transformer to classify seven facial emotions namely sad, neutral, anger, disgust, happiness, fear, and surprise. We have experimented using Vision Transformers, Spatial transformer + CNN (ST-CNN), and ResNet-50 on the FER2013 dataset. After multiple passes on different setups, we have obtained the best results from ST-CNN architecture with ReLU activation function, ADAM optimizer with a learning rate of 0.0001, batch size of 128, and L2 regularization with 0.01, which gave us the highest accuracy of 78.5%, a loss of 1.05 and F1-score 0.76.

Keywords: deep convolutional neural network, facial emotion recognition, image classification, deep learning, computer vision, and vision transformer.

b. Introduction

Facial expressions are key components of human communication because they help us interpret others' intentions. Humans analyze facial expressions and verbal tone to infer others' emotions, such as joy, sadness, and rage. Verbal components convey one-third of human communication, while nonverbal components convey two-thirds, according to various polls (Mehrabian, 1965; Kaulard et al., 2012). Facial expressions are one of the most important nonverbal components in interpersonal communication because they contain emotional significance. As a result, it's no surprise that facial emotion research has gotten a lot of attention in recent decades, with applications ranging from perceptual and cognitive sciences to affective computing and computer animations (Kaulard et al., 2012).

With the rapid development of artificial intelligence capabilities, there has been an increase in interest in automatic facial emotion recognition (FER). Although many sensors such as an electromyograph (EMG), electrocardiogram (ECG),electroencephalog

raph (EEG), and the camera can be used for FER inputs, the camera is the most promising since it gives the most informative hints for FER and does not require wearing.

This paper splits automatic FER research into two groups based on whether the features are created or generated by a deep neural network's output.

The FER is organized into three fundamental steps, as indicated in Fig. 1, (1) face and facial component identification, (2) feature extraction, and (3) expression categorization, in traditional FER approaches. First, a face image is extracted from an input image, followed by the detection of facial components (e.g., eyes and nose) or landmarks in the face region. Second, from the facial components, various spatial and temporal features are derived. Third, utilizing the retrieved features, pre-trained FE classifiers such as a vector support machine (SVM), AdaBoost, and random forest gives recognition results.

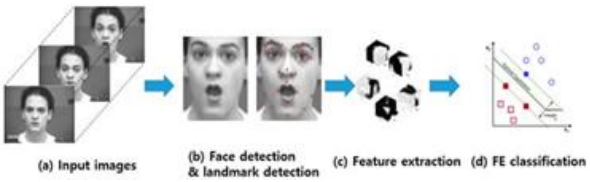

Figure 1 Procedure used in conventional FER approaches (Ko, 2018)

Deep learning has evolved as a universal technique for machine learning, delivering state-of-the-art outcomes in many computer vision studies with the availability of massive data (Ebrahimi Kahou et al., 2015), in contrast to older approaches employing handmade features.

By allowing "end-to-end" learning in the pipeline directly from the input photos, deep-learning-based FER approaches greatly minimize the dependence on face-physics-based models and other pre-processing techniques (Walecki, 2017). The convolutional neural network (CNN), a kind of deep learning, is the most popular network model among the various deep-learning models available. The input image is convolved through a filter collection in the convolution layers to build a feature map in CNN-based techniques. The result of the SoftMax method is used to merge each feature map into fully linked networks, and the facial expression is recognized as belonging to a specific class. The mechanism employed by CNN-based FER approaches is shown in Fig. 2.

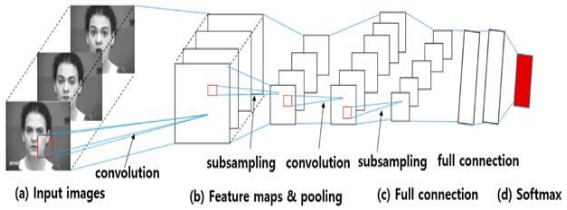

Figure 2 Procedure of CNN-based FER approaches (Ko, 2018)

This report presents two different techniques for facial emotion recognition and the experiments are tested on a well-known database called the FER2013 dataset (Facial emotion recognition 2013 dataset).

c. Problem Statement

- **Problem definition**

The goal of this research work is to experiment with different approaches and to find out differences in performances while targeting SOTA results for the FER challenge. We have also reviewed the research done on Facial Emotion Recognition (FER) as part of this project and we have presented a comprehensive analysis of our findings from our research.

- **Motivation**

To recognize and understand the intentions of a person emotions play an important role. There are generally seven raw emotions: anger, disgust, fear, happiness, neutral, sad, and surprise. Detecting the emotions needs to be very effective. When it comes to camera surveillance, for instance, we can identify the suspects based on fear/tension expressed by a person. The scope of this topic also assists in identifying expressions for normal people as well as people with mental disorders, such as schizophrenia, depression, etc.

- **Justification**

Considering its applications in vast areas like in the field of medicine that helps and assists psychologists, Security officers to identify criminals, etc., it is critical that we come up with a robust and efficient emotion recognition system that can provide utmost accuracy with high precision.

d. Literature Review

Since the late 1990s, CNNs have been developed and shown tremendous potential in image processing (Lecun et al., 1998). A typical CNN usually has three kinds of layers: a convolutional layer, a pooling layer, and a dense layer. By taking advantage of these three layers, although CNN is proficient in static image manipulation and recognition. the application of CNNs was constrained because of the lack of training data source and computing power. With the development of technology, CNNs are benefited from the growth of computing power and the increasing size of datasets. As the results, CNNs became a more reliable tool in feature extraction and image classification (Krizhevsky et al., 2017). Furthermore, numbers of techniques have been proposed to improve CNN performance. For example, Dahl et al. (2013) suggested that the alternative of Sigmoid function is Rectified Linear Unit (ReLU) activation, which can significantly avoid gradient dispersion problems and speed up training. Giusti et al. (2013) and Albawi et al. (2017) supported that different pooling methods such as average pooling and max pooling are used to down-sample the inputs and aid in generalization. To prevent overfitting, some techniques such as dropout and regularization are introduced. In order to increase accuracy and reduce the loss, different optimization algorithms have been developed and used in training. Sun (2019) suggested that though there is no systematic theoretical guideline on choosing an optimizer, empirical results show that a suitable optimization algorithm can effectively improve a model's performance. The most commonly used optimizer is Stochastic Gradient Descent (SGD). It is a simple technique that updates the parameters of a model based on the gradient of a single data point (2019). Apart from increasing the performance, variations of SGD have been proposed to speed up training. AdaGrad is one the example. It adaptively scales the learning rate for each dimension in the network (Lydia & Francis, 2019). Adam is another example that combines the advantages of AdaGrad and RMSProp by scaling the learning rate and introducing the momentum of gradient (Kingma & Ba, 2014). Learning rate is another main factor that can improve CNN performance. Larger learning rates may cause fluctuations around the minimum or divergences in the loss. Smaller learning rates can significantly slow down the model's convergence rate and may trap the model in non-optimal local minima. A commonly used technique is to employ a learning rate scheduler that changes the learning rate during training (Chin et al., 2015). For instance, time-based decay reduces the learning rate either linearly or exponentially as the iteration number increases. Step decay

drops the learning rate by a factor after certain epochs (Li & Arora, 2019).

Accompanied by the development of the performance of CNN, numerous of image datasets have been created aimed to train CNN models in different scenarios. In 2013, ICML introduced FER2013, which is one of the benchmarks for emotion recognition. Many CNN variants have achieved remarkable results with classification accuracy. For example, Liu et al. (2016) instructed an ensemble Deep Learning architecture that has three separate CNNs in order to improve overall performance. The best accuracy is 62.44 %. Minaee et al. (2021) developed an attentional convolutional network that achieved an accuracy of 70.02 %. Tang et al. (2015) introduced a deep neural network containing a support vector machine instead of a SoftMax layer and achieved an accuracy of 71.2 %. Shi et al. (2021) substituted the pooling layer with a Amend Representation Module (ARM) received a testing accuracy of 71.38 %. Pramerdorfer et al. (2016) compared the performance of three different architectures, VGG, Inception, and ResNet. Their results show that VGG performs best at an accuracy of 72.7 %, followed by ResNet at 72.4 %, and Inception at 71.6 %. Agrawal et al. (2019) made a study on the influence of variation of the CNN parameters on the recognition rate. By resizing the images to 64x64 pixels and turning the number of filters as well as the size of the kernel on a simple CNN, two novel models of CNN containing two successive convolution layers achieve an average accuracy of 65.23% and 65.77%.

The transformer is a classic NLP model proposed by Google's team in 2017, and Bert is also based on it. The transformer model uses the Self-Attention mechanism and does not use the sequential structure of RNN, which means the model can be trained in parallel and can have global information. Dosovitskiy et al. (2021) introduced Vision Transformer (ViT) that attains excellent results compared to state-of-the-art convolutional and requires fewer computational resources to train. Zhong and Deng (2021) showed the feasibility of Transformer models in face recognition and report promising experimental results.

e. Methodology

We conducted a three-way experiment, which includes ViT, ST-CNN, and ResNet-50. The high-level flow of our experiments is visualized in Fig. 3. In our experiments, we have tried different setups and tuned the hyperparameters in all the individual models. Finally, we have compared the results among the experiments and focused on the best result from each of the three models. We considered the best result based on the highest accuracy, F1 score and lowest loss.
Below sections provide a detailed description of each custom network built:

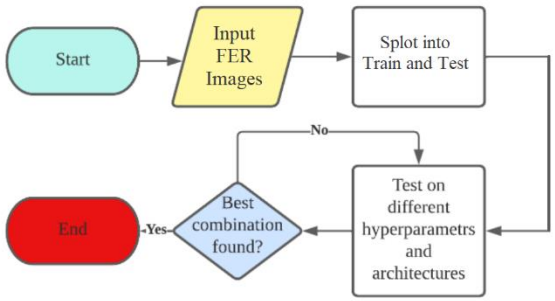

Figure 3 Flow chart of experiments

- ## Dataset

The dataset that has been chosen for this project is Fer2013 which has been collected from Kaggle competition. This dataset consists of approximately 35,887 grayscale

images of faces and the size of the images is 48 X 48. The data have been categorized into 7 classes: 0=Angry, 1=Disgust, 2=Fear, 3=Happy, 4=Sad, 5=Surprise, and 6=Neutral. The dataset has been split into two sets: the training set, and the test set. From Fig. 4 we can see the class distribution of the dataset. It is an imbalance dataset with having only 2% of the data in the disgust class and 25% in the happiness class. In the Fig. 5 the snapshot of the dataset is shown.

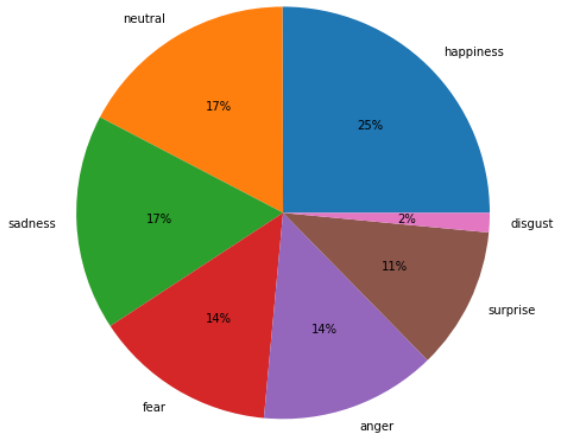

Figure 4 Classes distribution of dataset

Figure 5 Examples of FER2013 dataset

- **Models/Architectures**

ST-CNN

A spatial transformer network is a specialized type of CNN. It contains spatial transformer modules that attempt to make the network spatially invariant to its input data. Invariance is the ability of the model to recognize and identify features even when the input is transformed or slightly modified. In other words, a spatial transformer network is used when attempting to stabilize or clarify an object within a processed image or video. This leads to more accurate object classification and identification. The modulus of special transformers can be inserted into existing convolutional architectures. There are 3 main components of the spatial transformer as shown in Fig. 6. The first component is the localization network. This network takes a 4-dimensional tensor, that is (Width x Height x Channels x Batch Size), to represent a batch of images as input. It is a simple neural network with a few convolution layers and a few dense layers. It predicts the parameters of transformation as output, which is marked as Theta. These parameters determine the angle by which the input has to be rotated, the amount of translation to be done, and the scaling factor required to focus on the region of interest in the input images. Then, based on the parameters from localization, grid generator constructs a grid of coordinates in the input image corresponding to each pixel from the target image. The transformation technique involved here is Affine transformation. Finally, the sampler uses the parameters of the transformation and applies it to the input image using bilinear interpolation. In general, the input of spatial transformer is full images from datasets and the output is a transformed feature map.

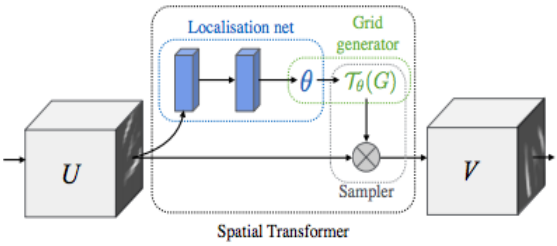

Figure 6 Spatial Transformer model (Jaderberg et al., 2016)

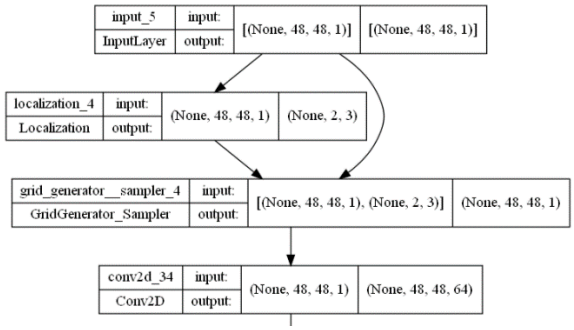

Figure 7 Flow chart about Spatial Transformer layers

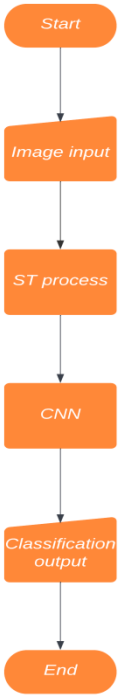

Figure 8 Flow chart about the entire process of classification

Fig. 7 and Fig. 8 display how the spatial transformer is incorporated into the deep CNN model in detail. In the localization layer, the image size is shrinking to 2x3 because it is the size of the affine transformation matrix. In the grid generator and sampler layer, both the input feature map and the sampling grid are processed by using, for example, bilinear interpolation to generate the transformed feature map, which has dimension 48x48. After that, deep CNN will be starting to classify the images and categorize them into the number of classes we designed.

Vision Transformer

Transformer (Vaswani et al., 2017) is a sequence-to-sequence architecture that heavily relies on self-attention techniques. The main advantage of transformer architecture is that it successfully overcame recursion and convolutional computations, making them superior to recurrent neural networks (RNN) and convolutional neural networks (CNN). Transformer made its highlight by a breakthrough result in natural language processing (NLP) tasks. After the success of NLP, it employed other domains as well. It achieved another benchmark result in computer vision (CV) tasks. Dosovitskiy et al. (2020) used large-scale pretraining and fine-tuning of a vanilla Vision Transformer to achieve state-of-the-art performance on picture classification datasets (He et al., 2022).

As transformers are seq-to-seq models, images can not be used as a direct input for this type of architecture. The architecture we have chosen, "The Vision Transformer" [ViT], solves this problem by splitting an image into smaller patches of a predefined size. The generated patches are then flattened into a linear embedding similar to tokens. So, the 2D images are turned into a 1D sequence of token embeddings. Once an image is broken into patches, we lose the structures of the input. For that reason, these tokens are then concatenated with learnable positional

embeddings, which helps retain the patches' positional information. In this stage, there is another unique token called the 'class' token, which is inspired by BERT. This class token is randomly generated; it does not provide helpful information alone. However, the class token gathers information from other tokens in the series with increasing depth and level of a transformer. When the ViT finally conducts the sequence's final classification, it employs an MLP head that only examines data from the last layer's Class Token and no other information.

The encoder of the transformer consists of the multiheaded self-attention (Vaswani et al., 2017), which is sandwiched between two layer normalization layers, multi-layered perceptron, and residual connection after every block (Wang et al., 2019). The Fig. below visualizes the ViT architecture.

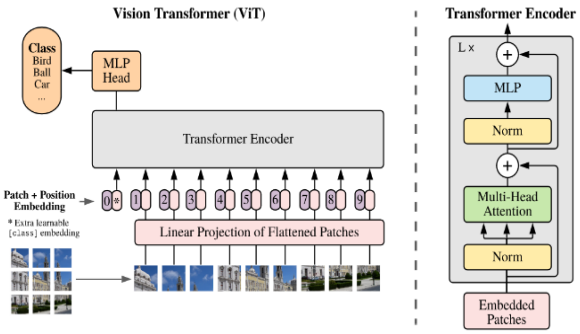

Figure 9 9Vision Transformer Architecture: (Dosovitskiy et. al., 2021)

ResNet-50

ResNet-50 model (Residual Network (ResNet-50) model is a Convolutional Neural Network (CNN) that has 50 layers. It is widely used in the area of Computer vision to solve problems related to image recognition, image classification, face recognition, object detection etc.

The ResNet-50 model consists of 5 stages each with multiple Identity blocks and a convolution block. Each convolution block has 3 convolution layers, and each identity block also has 3 convolution layers. The ResNet-50 has over 23 million trainable parameters. The use of 3-layer bottleneck blocks as well as 'Skip Connections' of this approach improves accuracy of model and reduce training time of data (He et al., 2016).

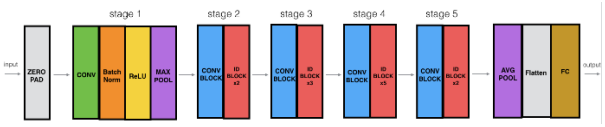

Figure 10 ResNet 50 Architecture (Dwivedi, 2019)

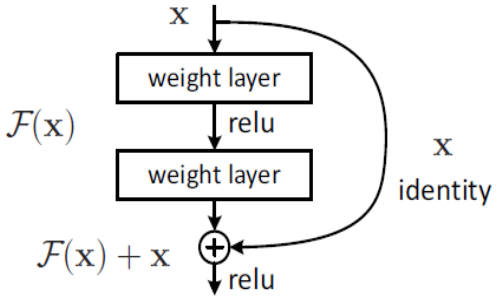

Figure 11 Skip Connections in ResNet 50 model

f. Experimental Setup

Our experiments were performed using the FER2013 dataset obtained through the Kaggle competition repository. This dataset's first column labelled 'emotion' was used as the target/dependent variable. In the second column, *pixels* hold the image in pixel values ranging from 0 to 255.0, representing the features (independent variables) for our image and when encoded can be seen as a grayscale image. FER2013 data set which consists of ~35800 images of size 48 x 48 where we have considered the train, and test split of 80:20. To construct the algorithms, we use Python3 IDE in Anaconda – Jupyter Notebook. This experiment is performed on

the 11th Gen Intel® Core Processor with GPU RTX 3060 and 16GB RAM running on the Windows Platform.

g. Results and Discussions

ST-CNN

Since ST-CNN model did not implement sequential using the Keras library, we are not able to use the hyperparameter tuners that Keras provided. We observed that when the learning rate is around 0.0001, the accuracy of the ST-CNN model is the highest and most stable. In this case, we decided to keep the learning rate as 0.0001 and turned the batch size to test how it would affect the overall performance.

We also compared the performance of ST-CNN and normal deep CNN. We found that ST-CNN has very little improvement in terms of both accuracy and loss compared to normal deep CNN. The reason is because the faces in the dataset have been automatically registered, which means the face is centered and occupies about the same amount of space in each image.

ST-CNN has very little improvement in terms of both accuracy and loss compared to normal DCNN.

Batch Size	Accuracy	Loss	AUC	F1
16	75%	1.27	0.94	0.72
32	74%	1.29	0.94	0.72
64	78%	1.12	0.95	0.75
128	78.5%	1.05	0.95	0.76

Table 1 Hyperparameters tuning results

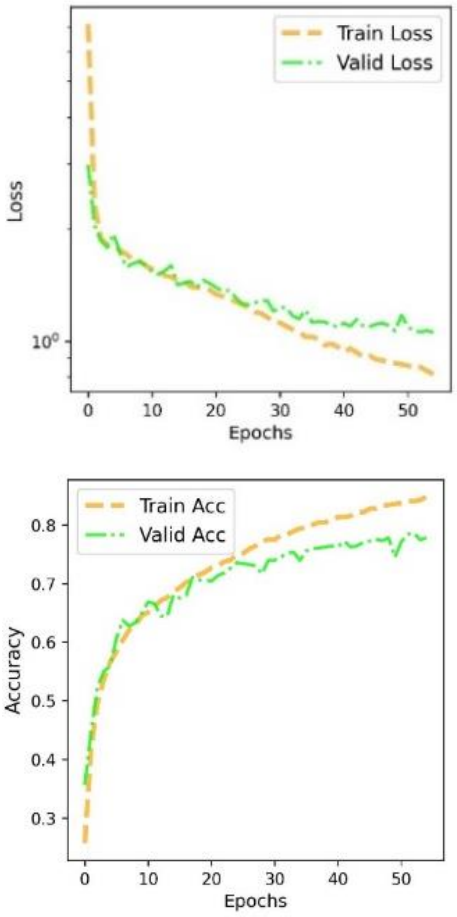

Figure 12 Accuracy and loss results of ST-CNN model

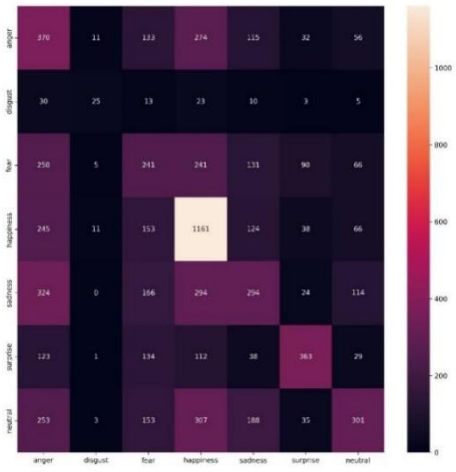

Figure 13 Heatmap of ST-CNN

ResNet50

To implement ResNet50 model, at first, we defined the identity blocks to transform the CNN into a residual network and built the convolution block. Then we built the 50-layer model by combining both identity and convolution blocks and we set the input shape as (48, 48, 1). Finally, we have trained our model to conduct the classification. We have evaluated our model on the test set and received 67.75% accuracy and 0.95 loss. Number of epochs = 50

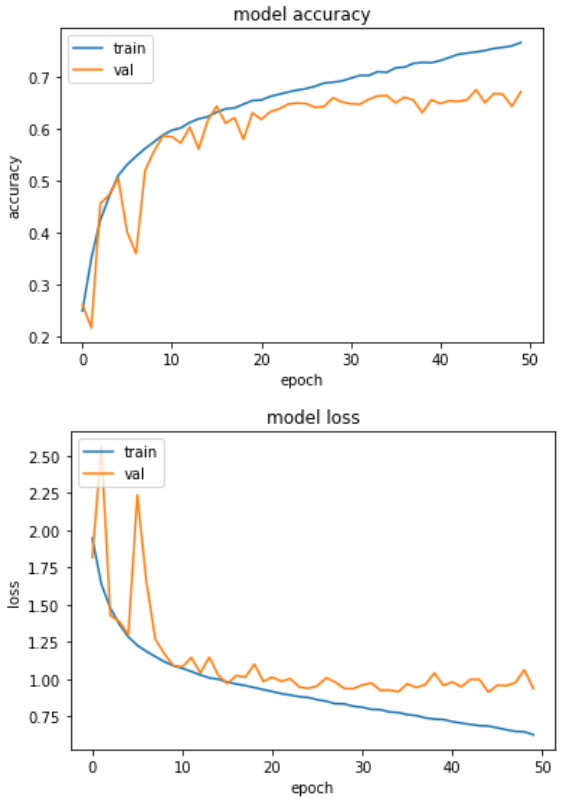

Figure 1412 Accuracy and loss results of ResNet 50 model

ViT

While training the ViT we have kept a few hyperparameter constants which include batch size=64, epochs=60, randomly generated positional embeddings with STD=0.02, multi-head attention dropout=0.1, two-layered MLP with dense blocks, GeLU activation and a dropout rate=0.1, optimizer=Adam. We made the learning rate dynamic by incorporating the ReduceLROnPlateau function of TensorFlow which monitors the loss of validation with a patience=5, factor=0.2, and lowest learning rate=1e-5. Our initial learning rate was 2e-3 for all the test cases. Furthermore, we also incorporated the EarlyStopping which monitors the accuracy of validation as we had noticed overfitting, and while training the model this metric gets stuck midway through the epochs. For that reason, we set that with a patience=10. These fixed hyperparameters have been referred from the original ViT paper (Dosovitskiy et al. (2020) several open-source implementations of ViT classifiers.

As for the other hyperparameters we experimented on the following Table 2 setup. We switched the kernel initializer from He_uniform to Zeros after running small tests and conducting literature reviews. From the experiments, it can be seen that even after increasing the complexity of the model the score keeps decreasing. We have identified four reasons for this lack of performance. First of all, the dataset size is quite small for a transformer. Secondly, the class imbalance in the dataset. Thirdly, we trained from the scratch without transfer learning. Last but not least is because of the small image size of 48X48.

Transformer Layers	Patch Size	Hidden Size	Num. of Heads	MLP dims	Kernel initializer	Accuracy	Loss	F1 score
6	16	64	4	128	He_uniform	39.27%	1.55	0.37
6	16	64	4	128	Zeros	40%	1.54	0.38
12	14	128	8	256	Zeros	26%	1.77	0.25
18	12	192	12	384	Zeros	25%	1.78	0.25

Table 2 ViT Hyper parameters tuning results

From the Fig. 15 below we can see the accuracy and the loss curves of ViT.

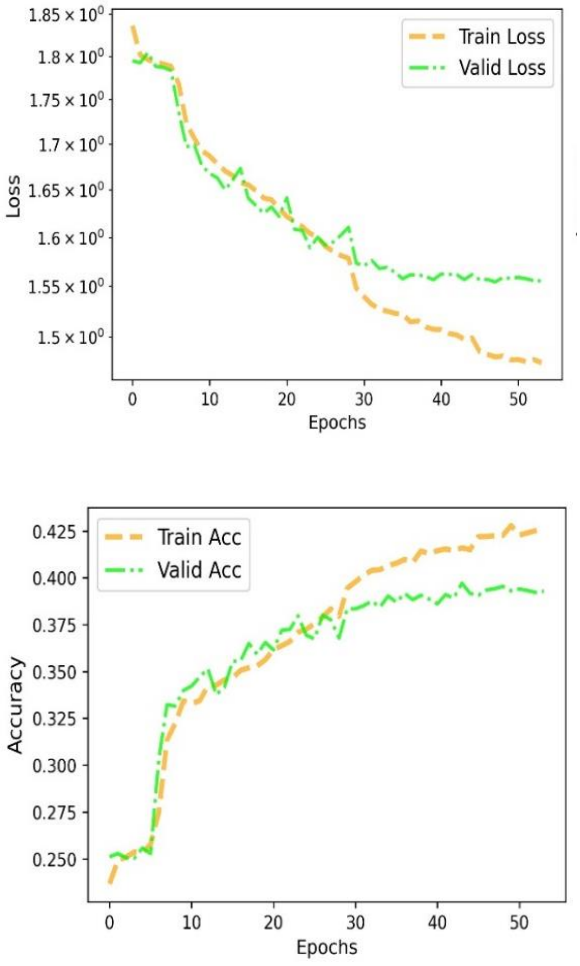

Figure 15 Loss & Accuracy of ViT

Fig. 16 visualizes the heatmap of class labels generated from the ViT.

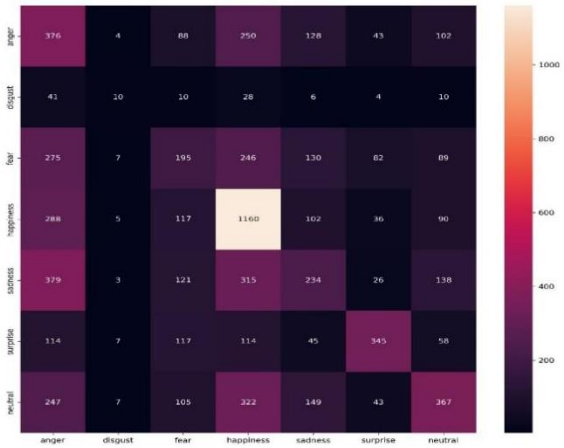

Figure 16 Heatmap of ViT

h. Conclusion

In this paper, we have experimented with the ViT and two different CNN architectures. From the ViT we can conclude that even with increasing the complexity of this structure it is difficult to obtain good results even after tuning the hyperparameters. This can be resolved by using a better dataset with a large number of data if we are training it from scratch or using transfer learning. As to ST-CNN, we noticed that the accuracy can be improved by increasing the batch size.

In terms of ResNet-50 while conducting the experiments we have achieved an accuracy of 67.75% which is lesser than the ST-CNN. However, the obtained loss is better than all the models that have been implemented.
From the results, we found our best result with ST-CNN architecture, ReLU activation function, Adam optimization function with a

Learning rate of 0.001, and L2 regularization with 0.01, which gave us the highest accuracy of 78.5, best F1 Score of 0.76 and a loss of 1.05.

i. Future Works

Till today we do not have any efficient architecture which can detect facial emotions proximate to 100%. Therefore, there are a lot of opportunities for improvement in designing a model which may include fusion-based methods for facial emotions recognition. Deep learning models require large datasets to train them to get better accuracy results but there is no large dataset available. Data augmentation can be applied to increase the size of the datasets as well as to avoid overfitting issues. The pre-trained models have produced state of art results in different areas; hence they should be used to train facial emotion detection systems. Finally, transfer learning techniques, which have a lot to offer in this domain, can also overcome major weaknesses with current emotion recognition systems.

j. References

Albawi, S., Mohammed, T. A., & Al-Zawi, S. (2017). Understanding of a convolutional neural network. *2017 International Conference on Engineering and Technology (ICET)*. https://doi.org/10.1109/icengtechnol.2017.8308186

Agrawal, A., & Mittal, N. (2019). Using CNN for Facial Expression Recognition: A study of the effects of kernel size and number of filters on accuracy. *The Visual Computer*, *36*(2), 405–412. https://doi.org/10.1007/s00371-019-01630-9

Chin, W.-S., Zhuang, Y., Juan, Y.-C., & Lin, C.-J. (2015). A learning-rate schedule for stochastic gradient methods to matrix factorization. *Advances in Knowledge Discovery and Data Mining*, 442–455. https://doi.org/10.1007/978-3-319-18038-0_35

Dahl, G. E., Sainath, T. N., & Hinton, G. E. (2013). Improving deep neural networks for LVCSR using rectified linear units and dropout. *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*. https://doi.org/10.1109/icassp.2013.6639346

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., & Houlsby, N. (2021, June 3). *An image is worth 16x16 words: Transformers for image recognition at scale*. arXiv.org. Retrieved April 26, 2022, from https://arxiv.org/abs/2010.11929

Dwivedi, P. (2019). *Understanding and Coding a ResNet in Keras*. https://towardsdatascience.com/. Retrieved 26 April 2022, from https://towardsdatascience.com/understanding-and-coding-a-resnet-in-keras-446d7ff84d33.

Ebrahimi Kahou, S., Michalski, V., Konda, K., Memisevic, R., & Pal, C. (2015). Recurrent Neural Networks for Emotion Recognition in Video. *Proceedings Of The 2015 ACM On International Conference On Multimodal Interaction*. https://doi.org/10.1145/2818346.2830596

He, X., Li, C., Zhang, P., Yang, J., & Wang, X. E. (2022). Parameter-efficient Fine-tuning for Vision Transformers. *arXiv preprint arXiv:2203.16329*.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770-778).

Giusti, A., Ciresan, D. C., Masci, J., Gambardella, L. M., & Schmidhuber, J. (2013). Fast image scanning with deep max-pooling Convolutional Neural Networks. *2013 IEEE International Conference on Image Processing*. https://doi.org/10.1109/icip.2013.6738831

Jaderberg, M., Simonyan, K., Zisserman, A., & Kavukcuoglu, K. (2016, February 4). *Spatial Transformer Networks*. arXiv.org. Retrieved April 26, 2022, from https://arxiv.org/abs/1506.02025

Kaulard, K., Cunningham, D., Bülthoff, H., & Wallraven, C. (2012). The MPI Facial Expression Database — A Validated Database of Emotional and Conversational Facial Expressions. *Plos ONE*, *7*(3), e32321. https://doi.org/10.1371/journal.pone.0032321

Kingma, D. P., & Ba, J. L. (2014). Adam: A method for stochastic optimization. *International Conference on Learning Representations*.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2017). ImageNet classification with deep convolutional Neural Networks. *Communications of the ACM*, *60*(6), 84–90. https://doi.org/10.1145/3065386

Ko, B. (2018). A Brief Review of Facial Emotion Recognition Based on Visual Information. Sensors, 18(2), 401. https://doi.org/10.3390/s18020401

Lecun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11), 2278–2324. https://doi.org/10.1109/5.726791

Li, Z., & Arora, S. (2019). An Exponential Learning Rate Schedule For Deep Learning.

Liu, K., Zhang, M., & Pan, Z. (2016). Facial expression recognition with CNN ensemble. *2016 International Conference on Cyberworlds (CW)*. https://doi.org/10.1109/cw.2016.34

Lydia, A. A., & Francis, F. S. (2019). Adagrad - An Optimizer for Stochastic Gradient. *International Journal Of Information And Computing Science*, *6*(5).

Mehrabian, A. (2008). Communication without words. *Communication theory*, *6*, 193-200.

Minaee, S., Minaei, M., & Abdolrashidi, A. (2021). Deep-emotion: Facial expression recognition using attentional convolutional network. *Sensors*, *21*(9), 3046. https://doi.org/10.3390/s21093046

Pramerdorfer, C., & Kampel, M. (2016, December 9). *Facial expression recognition using convolutional neural networks: State of the art*. arXiv.org. Retrieved April 26, 2022, from https://arxiv.org/abs/1612.02903v1

Sun, R. (2019, December 19). *Optimization for deep learning: Theory and algorithms*. arXiv.org. Retrieved April 26, 2022, from https://arxiv.org/abs/1912.08957v1

Shi, J., Zhu, S., & Liang, Z. (2021, October 11). *Learning to amend facial expression representation via de-albino and affinity*. arXiv.org. Retrieved April 26, 2022, from https://arxiv.org/abs/2103.10189

Tang, Y. (2015, February 21). *Deep learning using linear support vector machines*. arXiv.org. Retrieved April 26, 2022, from https://doi.org/10.48550/arXiv.1306.0239

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, *30*.

Walecki, R., Rudovic, O., Pavlovic, V., Schuller, B., & Pantic, M. (2017). Deep structured learning for facial expression intensity estimation. Image Vis. Comput, 259, 143-154.

Wang, Q., Li, B., Xiao, T., Zhu, J., Li, C., Wong, D. F., & Chao, L. S. (2019). Learning deep transformer models for machine translation. *arXiv preprint arXiv:1906.01787*.

Zhong, Y., & Deng, W. (2021, April 13). *Face transformer for recognition*. arXiv.org. Retrieved April 26, 2022, from https://arxiv.org/abs/2103.14803