Facial Emotion Recognition using Deep Learning

Lian Duan
School of Computer Science
University of Windsor
Windsor, Canada
SID: 110058821

Naga Jyothirmayee Dodda
School of Computer Science
University of Windsor
Windsor, Canada
SID: 110083327

Roisul Islam Rumi
School of Computer Science
University of Windsor
Windsor, Canada
SID: 110080808

Sumaiya Deen Muhammad
School of Computer Science
University of Windsor
Windsor, Canada
SID: 110059602

***Abstract*—Human emotion recognition is one of the most sought-after topics in the computer vision field, especially in the facial feature extraction domain. Moreover, in this era of deep learning, much progress has been made in terms of performance metrics. Utilizing the power of modern technologies in this project, we have experimented with different types of classifiers which classify three strong emotions of humans. The emotions are positive, negative, and neutral. We have used the dataset from Facial Emotion Recognition Challenge (FERC), and feature engineered to convert the seven classes of the dataset to our custom three classes. In our experiments, we have used the vanilla Vision Transformer (ViT), Spatial Transformer + Convolutional Neural Network (ST-CNN), ResNet-50, and Semi-Supervised GAN (SGAN). After experimenting with a combination of different hyperparameters we found that ST-CNN gave us the best results with 83.9% for accuracy, 0.68 for loss, and 0.93 for F1 score.**

***Index Terms*—Facial Emotion Recognition, Deep Learning, Neural Network, Transformer, Generative Adversarial Network, Convolutional Neural Network, Image Classification, Computer Vision.**

I. Introduction

Facial expressions are critical for human communication because they assist us in interpreting the intentions of others. Humans use facial expressions and vocal tone to infer emotions like joy, sadness, and anger from others. According to different surveys, verbal components reflect one-third of human communication, whereas nonverbal components express two-thirds (Mehrabian, 1965; Kaulard et al., 2012). Because they carry emotional content, facial expressions are one of the most significant non-verbal components of interpersonal communication. As a result, it's no wonder that research into facial expression recognition has gained a lot of momentum in recent decades, with applications spanning from perceptual and cognitive sciences to affective computing and computer animations (Kaulard et al., 2012).

With the rapid advancement of artificial intelligence capabilities, interest in automatic face expression recognition has emerged (FER). Although several sensors can be utilized for FER inputs, including an electromyograph (EMG), electrocardiogram (ECG), electroencephalograph (EEG), and the camera, the camera is the most promising since it provides the most promising information through hints for FER.

Automatic FER research can be broadly categorized into two divisions, depending on whether the features are constructed or generated by the output of a deep neural network.

In conventional FER techniques, the FER is categorized has three core processing stages, as shown in Figure 1, (1) face and facial component identification, (2) feature extraction, and (3) expression categorization apart from input. After extracting a face image from an input image, facial components (e.g., eyes and nose) and landmarks in the face region are detected. Second, numerous spatial and temporal features are obtained from the facial components. Third, pre-trained classifiers such as a support vector machine (SVM), AdaBoost, and random forests use the extracted features to generate recognition results.

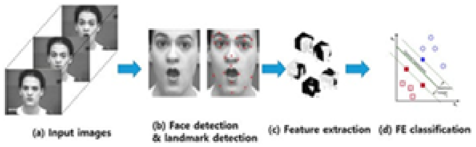


Fig. 1. Procedure used in conventional FER approaches (Ko, 2018)

Deep learning has emerged as a ubiquitous machine learning technique, giving cutting-edge results in many computer vision research with the availability of enormous data (Ebrahimi Kahou et al., 2015), in contrast to prior approaches that used handcrafted features. Deep-learning-based FER methods considerably reduce the need on face-physics-based models and other pre-processing techniques by facilitating ”end-to-end” learning in the pipeline straight from the input pictures (Walecki, 2017). Among several deep-learning models available, the CNN is the most used model. In CNN-based approaches, the input picture is convolved via a filter collection in the convolution layers to generate feature maps. The generated feature maps are then flattened and passed through a SoftMax layer for classification. Figure 2. Visualizes the workings of a CNN-based model below.

This report represents the findings and test results obtained by experimenting on a well-known dataset called the FER2013 dataset, using three distinct algorithms for facial emotion detection.

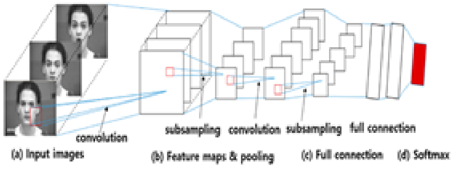


Fig. 2. Procedure of CNN-based FER approaches (Ko, 2018)

II. problem Statement

A. Problem Definition

The objective of this research is to test various approaches and see if there are any changes in performance while focusing on state-of-the-art for the FERC. In addition, as part of this effort, we analyzed the literature on FER and offered a detailed analysis of our findings.

B. Motivation

Emotions play a significant role in recognizing and understanding a person's intentions. Anger, disgust, fear, happiness, neutral, sad, and surprise are the seven basic raw emotions. Emotion detection must be extremely precise. When it comes to video surveillance, for example, we can identify suspects based on a person's anger, nervousness, or anxiety. The vastness of this area also aids in detecting expressions of both normal person's and those suffering from mental illnesses such as schizophrenia and sadness which helps assist in the field of medicine. A good FER classifier can also be used as an assistive system for a psychiatrist to deal with different types of patients.

C. Justification

Given its wide range of applications, including helping and assisting psychologists, security personnel in investigating crimes, and so on, it is vital that we develop a strong and efficient emotion detection system that can achieve optimum accuracy and precision.

III. 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.

IV. Methodology

A. Material and Data

FER2013, which was acquired through the Kaggle competition, was used as the dataset for this research. The dataset contains csv values of pixels which can be converted into 35,887 grayscale photos of facial emotions, each measuring 48 X 48 pixels. The data is divided into seven categories: 0=Angry, 1=Disgust, 2=Fear, 3=Happiness, 4=Sadness, 5=Surprise, and 6=Neutral.

From different studies, we decided to pre-process the data with feature engineering where we cluster several emotions together. For the clusters we have focused on strong emotions which can be broadly categorized into neutral, positive and negative segments (Aldahdouh et al., 2020). We have considered angry, disgust, dear, and sadness as negative emotions and represented this class with 0. In the rest of the seven emotions mentioned, we have grouped happiness and surprise as the positive emotions and depicted this emotion with label 1 and in the third group we used neutral and set 2 as its class label. The dataset's class distribution can be shown in Figure 3, and it can be said that it's an unbalanced dataset, with only 36% of the data in the positive emotion category, 17% in the neutral emotion category, and the rest of the 47% in the negative emotion category. A snapshot of the dataset is provided in the Figure 4.

B. Proposed Method

Our objective was to experiment with different types of architecture that are well known for image classification tasks. For that reason we used ViT, ST-CNN, SGAN and ResNet-50. Our dataset The high-level flow of our experiments is visualized in Figure 5. 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 diagram depicts the overall follow of our project built:

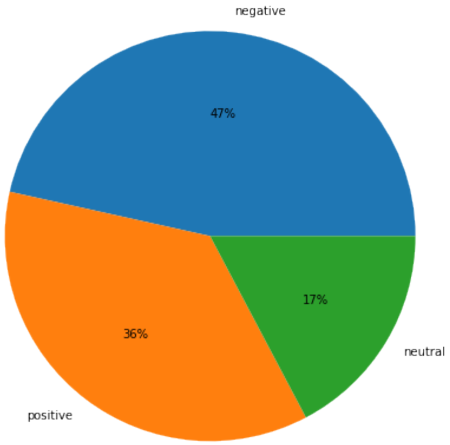


Fig. 3. Classes distribution of dataset

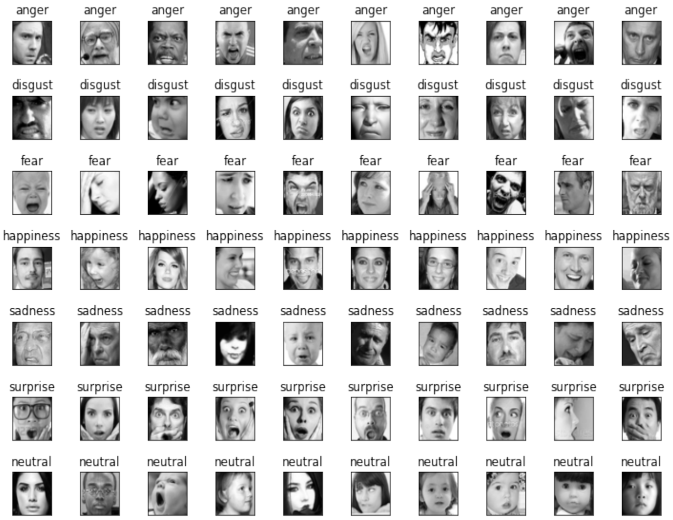


Fig. 4. Examples of FER2013 dataset

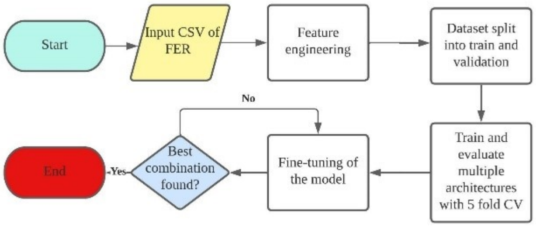

Start
Input CSV of FER
Feature engineering
Dataset split into train and validation
Train and evaluate multiple architectures with 5 fold CV
Fine-tuning of the model
Best combination found?
No
Yes
End

Fig. 5. Flow chart of Proposed methodological

C. Conditions and Assumptions

For our experiments we have grouped all the categories into neutral, positive and negative segments.Based on the study (Watson et al., 1988) we have grouped all these seven raw emotions into three categories. Anger, disgust, fear, and sadness are grouped as negative emotions. surprise and happiness are grouped as positive emotions and we have kept the neutral emotion as it is.

D. Simulation Analysis

We have converted all input data from CSV to images of size 48*48 dim. We have merged all the seven classes of emotions available into three categories, in order to identify the positivity and negativity of the emotion(Positive, Negative and Neutral).Once the data is set we have Split dataset into train and test sets using train_test_split.A batch size of 16 or 32 is recommended, so we used 32 in this experiment.We have Measured our performance on the Test set.

V. Computational Experiments

A. Experiments and Experimental setup

We used the FER2013 dataset from the Kaggle Repository in our research. The target/dependent variable was picked from the dataset's first column, which is labelled as 'emotion.' In the second column, pixels store the images in pixel values format ranging from 0 to 255.0, representing the image's attributes (independent variables), and when encoded, these pixels produce grayscale facial images. We utilized the FER2013 data set, which comprises roughly 35800 photographs with a size of 48 by 48 pixels and we split it in train-to-test with 80:20 ratio.

We used the Python 3.9, Jupyter Notebook along with Google Collab to build the algorithms. These experiments were run on a Windows platform with an 11th Gen Intel® Core i7 Processor, RTX 3060 GPU, and 16GB RAM.

Methodologies and libraries used in this project:

1) Datasets from CSV files, doing some basic introspection, iamge conversion using openCV2 , and Image tokenization was done using transformers,Image generation and classification is done using SGAN.
2) Splitting data into train and test splits for Machine Learning.
3) Applying all the models opted on FER2013 dataset.
4) Training and evaluating the model using TensorFlow , Keras, Pandas and scikit-learn.

B. Evaluation metrics

Accuracy: Accuracy is the ratio of correct predictions to total predictions. An accuracy metric is used to measure the algorithm's performance in an interpretable way and is usually determined after the model parameters and is calculated in the form of a percentage. It is the measure of how accurate the model's prediction is compared to the true data.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{1}$$

Loss: A loss function is used to optimize a machine learning algorithm. The loss is calculated on training and validation, and its interpretation is based on how well the model is doing in these two sets. It is the sum of errors made for each example in training or validation sets. Loss value implies how poorly or well a model behaves after each iteration of optimization. As for the loss function, Cross entropy is used for evaluation in this project.

As the final part of our evaluation, the model was checked against the test set. The data required for this was created, and the model's 'evaluate' method in TensorFlow was used to compute the metric values for the trained model. This method returns the loss value and metrics values for the model in test mode.

F1 Score: F1 score can be defined as the harmonic mean of the model's precision and recall. It is used to measure the rate of performance of a model.

$$F1Score = \frac{TP}{TP + 1/2(F + FN)} \tag{2}$$

Below are the abbreviations for the Formula

1) TP: True Positives
2) TN: True Negatives
3) FP : False Positives
4) FN: False Negative

C. Implementation Details

1) ST-CNN: Spatial Transformer Network (STN) is a specialized type of CNN. STN includes a number of spatial transformer modules that allow the model to recognize and classify features using data that is invariant to its input. Even when the input is marginally changed, invariance can aid the model in recognizing and identifying features. In other words, STN can increase overall performance while attempting to stabilize or clarify an object within a processed image or video. As a result, object classification and identification become more precise. Special transformer module can be introduced into convolutional networks that already exist. The spatial transformer has three primary components, as shown in Figure 6. The localization network is the initial component. To represent a set of photos as input, localization has width, height, channels, and batch size. Localization is a simple neural network consisting of few convolution layers and a few dense layers. Localization predicts the parameters of transformation as output, which is marked as Theta in Figure 6. These parameters determine several properties including rotation angle and scaling factor of the region of interest in the input images. Then, based on the parameters from the localization procedure, the 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 employs bilinear interpolation to apply the transformation's parameters to the input image. In general, the input of a spatial transformer is a collection of images from datasets, and the output is a transformed feature map.

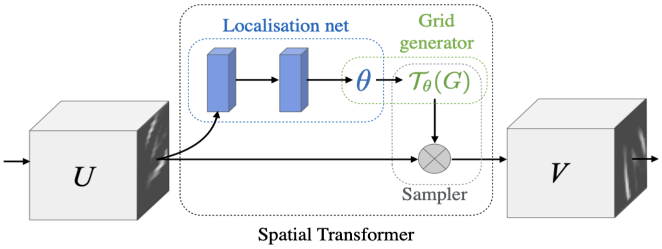
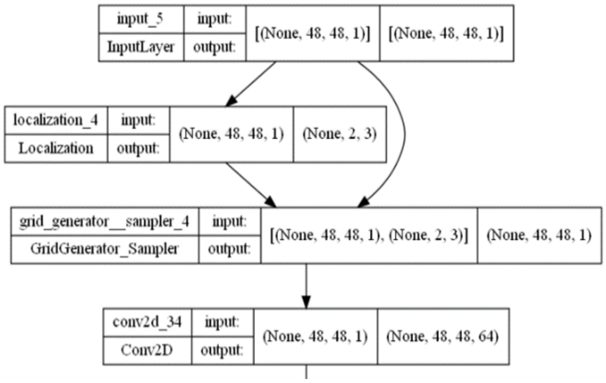


Fig. 6. Spatial Transformer model (Jaderberg et al., 2016)

The spatial transformer is included into the deep CNN model in detail in Figure 7 and Figure 8. To match the size of the affine transformation matrix, the picture size in the localization layer is reduced to 2x3. Both the input feature map and the sampling grid are processed in the grid generator and sampler layer, for example, using bilinear interpolation to generate a transformed feature map with a 48x48 size. Deep CNN next begins to classify and categorize the photos into the number of classes we specified.



input_5 InputLayer	input: output:	[(None, 48, 48, 1)]	[(None, 48, 48, 1)]
localization_4 Localization	input: output:	(None, 48, 48, 1)	(None, 2, 3)
grid_generator__sampler_4 GridGenerator_Sampler	input: output:	[(None, 48, 48, 1), (None, 2, 3)]	(None, 48, 48, 1)
conv2d_34 Conv2D	input: output:	(None, 48, 48, 1)	(None, 48, 48, 64)

Fig. 7. Flow chart about Spatial Transformer layers

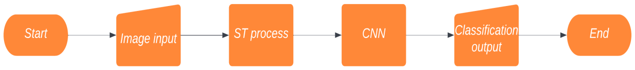


Fig. 8. Flow chart about the entire process of classification

2) ViT: Transformer (Vaswani et al., 2017) is a self-attention-based sequence-to-sequence architecture. The fundamental benefit of the transformer model is that it effectively overcame recursion and convolution computations, making it superior to recurrent neural networks (RNN) and CNN. Transformer earned a name for itself by achieving ground-breaking results in natural language processing (NLP) problems (Vaswani et al., 2017). Following its success in NLP, it was applied to additional areas, and among them, it obtained yet another benchmark performance in computer vision (CV) tasks. (Dosovitskiy et al., 2020) achieved state-of-the-art performance on image datasets using large-scale pretraining and fine-tuning of a vanilla Vision Transformer (He et al., 2022). Images cannot be utilized as a direct input for transformers as they are seq-to-seq models. ViT overcomes this problem by dividing a picture into smaller portions of a predetermined size. The produced patches are then flattened into a token-like linear embedding. As a result, the 2D pictures are converted into a 1D series of token embeddings. These tokens are then concatenated with learnable positional embeddings, which aids in the retention of patch positional information. There is another special token at this stage called the ’class’ token, which is influenced by Bidirectional Encoder Representations from Transformers (BERT) (Devlin et al., 2018). This class token is produced at random and does not give any relevant information on its own. The class token, on the other hand, acquires information from other tokens in the series as the depth and level of a transformer increase. When the Vision Transformer finally performs the sequence’s final classification, it utilizes an MLP head that does not use any other information except looking at data from the last layer’s class token. The transformer’s encoder is composed of a multiheaded self-attention (Vaswani et al., 2017) sandwiched between two-layer normalizing layers, a multi-layered perceptron, and a residual connection after each block (Wang et al., 2019). The ViT architecture is seen in Figure 9 below.

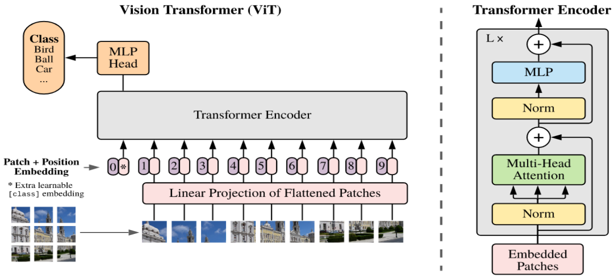


Fig. 9. Vision Transformer Architecture: (Dosovitskiy et. al., 2021)

3) 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 object detection, image classification, face recognition, image recognition, etc. The ResNet-50 model consists of 5 stages each with multiple Identity blocks and a convolution block which have been shown in Figure 10. Each convolution block has 3 convolution layers, and each identity block also has 3 convolution layers. The 3-layer bottleneck blocks as well as ‘Skip Connections’ of this approach, shown in Figure 11, betters the accuracy of model and reduce training time of data (He et al., 2016; Dwivedi, 2019).

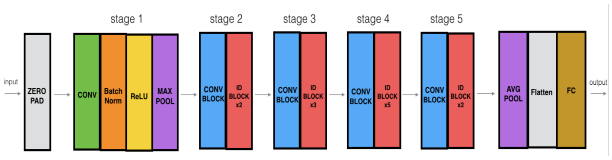


Fig. 10. ResNet-50 Architecture (Dwivedi, 2019)

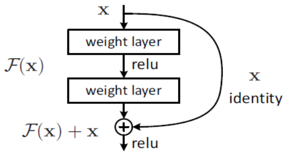

x
weight layer
relu
weight layer
$\mathcal{F}(\mathbf{x})$
x
identity
$\mathcal{F}(\mathbf{x}) + \mathbf{x}$
relu

Fig. 11. Skip Connections

4) SGAN: Generative Adversarial Networks (GAN) is a strong deep learning model in which two neural networks (generator and discriminator) fight to become increasingly accurate in their predictions to distinguish between real and fraudulent data. In GAN, A random noise vector is fed to generator 'g' as input which converts it into fake data and then sends it to the discriminator to test its identity. The discriminator 'd' is trained in an unsupervised manner to classify an image as real or fake, which is a binary classification. we focus on training the generator and discarding the discriminator history in this model.

SGAN is one of the enhanced approaches of GAN where it is designed by enhancing the discriminator network to output class labels, to form the semi-supervised setting (Salimans et al., 2016) On a dataset with inputs from one of k classes, we train a generative model g and a discriminator d. d is trained to predict which of k+1 classes the input belongs to, with an additional class added to match to g's outputs. It can be demonstrated that this method may be utilized to produce a more data-efficient classifier and that it can generate higher-quality samples than a traditional GAN. Figure 12 depicts the SGAN Architecture.

In Semi-Supervised GAN discriminator behaves like a feature extractor and we have extended it to both unsupervised and supervised classification. Unsupervised classification discriminator performs its function just like a normal binary classifier and helps strengthen the generator by sending the feedback to the generator. On the other hand, the supervised, Multiclass classification discriminator classifies and returns Multiclass discrimination loss to itself. we discard the generator history and keep the supervised classifier (discriminator) as our goal is to train a classifier with limited labeled data. In a normal GAN, the discriminator network d outputs a probability that the input image is drawn from the data generating distribution. Traditionally, a feed-forward network with a single sigmoid unit is used, but it can alternatively be done with a softmax output layer with one unit for each of the classes [REAL, FAKE]. In SGAN we made use of the following custom activation function for unsupervised discriminator:

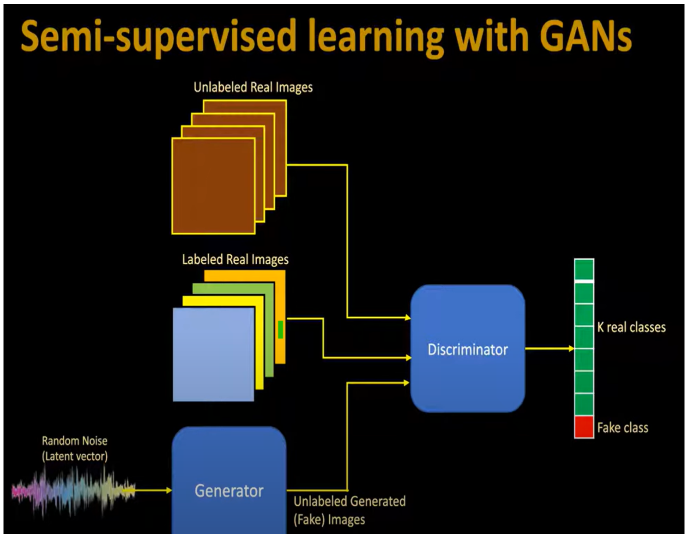


Fig. 12. SGAN Architecture

$$D(x) = \frac{Z(x)}{Z(x)+1} \tag{3}$$

Where Z(x):

$$Z(x) = \sum_{k=1}^{K} = \exp\left[l_{\mathrm{k}}(x)\right] \tag{4}$$

In the formulae mentioned above:

1) $\mathrm{l_k}$ is the supervised loss function of K classes of the original Classifier.
2) D(x) is the discriminator

As per the Paper (Salimans et al., 2016). It's easy to see how d may have k+1 output units matching to (Real classes (Class-1, Class-2,..., Class-k-1, Class-K), FAKE) after this modification is done. In this situation, d can also take the role of c. This network is referred to as d/c. It's analogous to training a GAN to train an SGAN. For the half of the minibatch that was taken from the data generating distribution, we simply apply higher granularity labels. d/c is trained to minimize the negative log-likelihood with respect to the given labels, whereas g is trained to maximize it

D. Results and Analysis

1) ST-CNN : Since the Sequential library of Keras does not apply to ST-CNN model, we can not use the hyperparameter tuners that Keras provided. We observed that when the batch size is 64, the accuracy of the ST-CNN model is the highest and most stable. In this case, we decided to keep the batch size as 64 and perform 5-fold Cross Validation to better evaluate the model.

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 deep CNN. The experimental results of ST-CNN can be seen in Table I. The accuracy and loss are shown in Figure 13 and Figure 14

TABLE I
ST-CNN TUNING RESULTS

5-fold Cross Validation	Batch Size	Accuracy	Loss	AUC	F1
1	64	83.6%	0.72	0.94	0.93
2	64	83.2%	0.72	0.94	0.93
3	64	83.9%	0.68	0.95	0.93
4	64	82.5%	0.73	0.92	0.93
5	64	82.8%	0.72	0.94	0.93

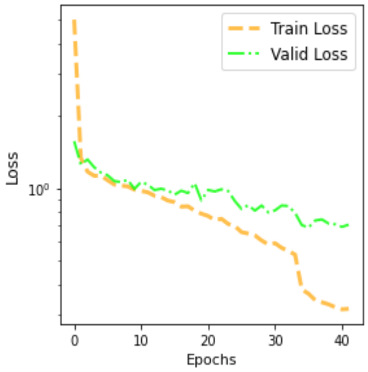


Fig. 13. Loss of ST-CNN

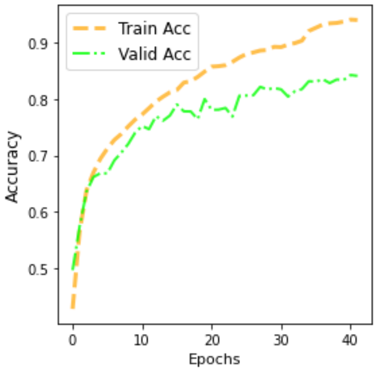


Fig. 14. Accuracy of ST-CNN

2) *RESNET-50:* To build ResNet-50 model, at the start, we have created the identity blocks to transform the CNN into a residual network and have built the convolution block. Then we built the 50-layer model by combining both identity and convolution blocks and we have set the input shape as (48, 48, 1). Finally, we have trained our model to perform the classification. We have evaluated our model on the test set and received 67.75% accuracy and 0.95 loss. The accuracy and loss graphs are presented in Figure 15 and Figure 16.

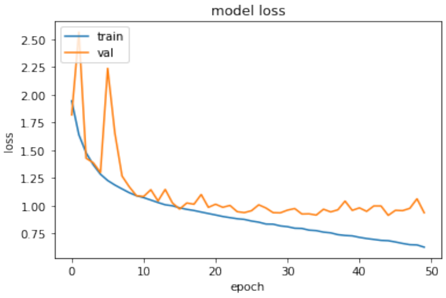


Fig. 15. ResNet-50 Accuracy

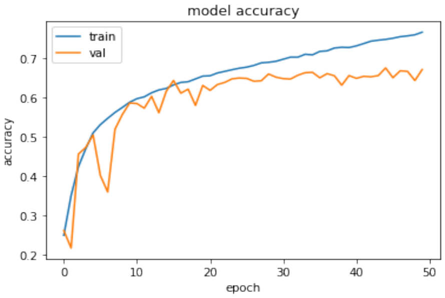


Fig. 16. ResNet-50 Loss

3) *ViT :* While training the ViT we have kept a few hyperparameter constants which include batch size=32, epochs=30, randomly generated positional embeddings with STD=0.02, multi-head attention dropout=0.1, four-layered MLP with two dense blocks, ReLU activation and two dropout layers with a dropout rate=0.1, and Adam optimizer. Moreover, we tried with six-layered MLP with three blocks each of dense and dropout layer. However, it was overfitting, and gave the lowest result which made us drop this idea. We also used GeLu (Hendrycks et al., 2016), but found ReLU was performing better. For that reason we made ReLU static on other experiments.

For learning rate we used Tensorflow's ReduceLROnPlateau function that monitors the test accuracy with a patience=2, factor=0.3, and lowest learning rate=1e-5. Our initial learning rate was 2e-3 for all the experiments. Furthermore, we integrated the EarlyStopping, which measures the accuracy of test since we had seen overfitting, and this metric becomes stuck midway through the epochs during training the model. As a result, we set patience=5 for it. Some of these predefined hyperparameters were mentioned in the original ViT publication (Dosovitskiy et al. (2020)) as well as on many open-source ViT classifier implementations.

TABLE II
ViT TUNING RESULTS

Transformer layers	Patch Size	Hidden Size	Num heads	MLP dims	Kernel init	Activation func	Accuracy	Loss	F1
4	16	64	4	128	he uniform	relu	54.81%	0.94	0.54
6	4	64	4	128	he uniform	relu	51.84%	0.97	0.49
6	10	64	4	128	he uniform	relu	49.61%	0.99	0.49
6	16	64	4	128	he uniform	relu	54.39%	0.94	0.54
6	16	64	4	128	he uniform	gelu	55.91%	0.92	0.55
6	16	64	4	128	zeros	relu	55.15%	0.93	0.55
10	16	64	4	128	he uniform	relu	52.83%	0.96	0.53
10	16	128	4	256	he uniform	relu	50.05%	0.99	0.49
12	16	64	8	256	he uniform	relu	46.53%	1.03	0.94

We tried with the other hyperparameters on the Table II configuration. After running short tests and undertaking literature survey, we changed the kernel initializer from zeros to he_uniform. According to the results of the experiments, the score continues to decrease as the model's complexity increases, it starts overfitting. Another element we observed from the experiment was the patch size. We tried by reducing it 6 units, but the results were dropping with the lowest one at patch size 10. We have determined five causes of this poor performance. To begin with, the dataset size for a transformer is relatively tiny. Secondly, class imbalance of the dataset. Thirdly, we trained from the ground up, with no transfer learning. Fourthly, lack of resource power for which we had to limit our tests on several factors like changing one hyperparameter at one time. Finally, the small picture size of 48X48.

From the Figure 17 below we can see the accuracy and the loss curves of ViT.

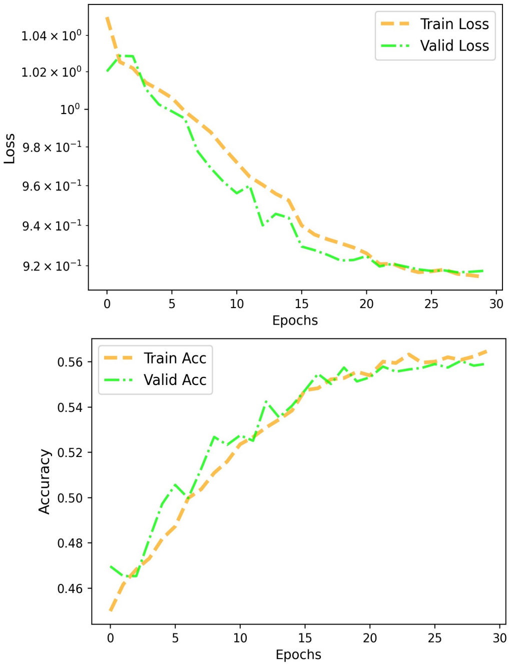


Fig. 17. Loss and accuracy of ViT

From the Figure 18 below we can see the heatmap of the classes.

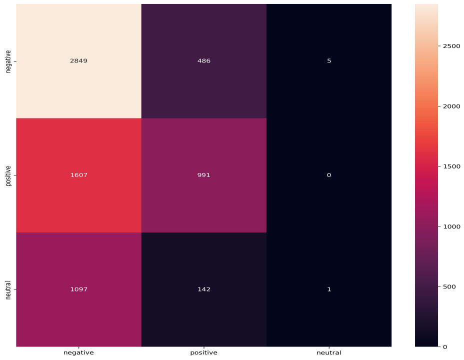


Fig. 18. Heatmap of ViT

4) SGAN: With respect to SGAN we have considered only two performance metrics i.e Accuracy and Loss. While training the discriminator network we have set the trainable status to false for an unsupervised setting as it the network should not retain the history and true for a supervised setting. we have considered a dropout value of 0.1-0.4 range with 0.1 scaling during the training process where we have obtained better results using the dropout value of 0.4. we also have experimented with two Loss functions "sparse_ categorical cross-entropy and categorical cross-entropy" in which sparse_categorical cross-entropy has better results. The results in the images from Figure 19-Figure 22 depict the accuracy and loss curves of the supervised discriminator during the training and testing.

Considering the parameters set on the network, we observed that dropout helps in improving the model performance as well as helps in increasing the complexity of the network by using this layer. The test accuracy obtained for the SGAN is very low which is approximately 38% which as been improved by 17% when hyper tuned. We have identified that there are possibly three reasons for the low performance. First of all, the images are in grayscale mode and hence discriminator might not be able to identify features of the faces in the images. Secondly, the small image size of 48X48 and imbalanced classes. Finally,

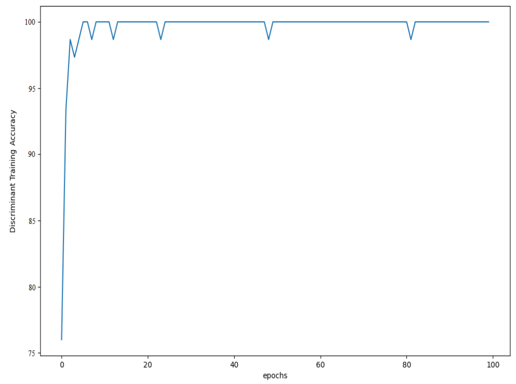


Fig. 19. Supervised Discriminator Training Accuracy Results

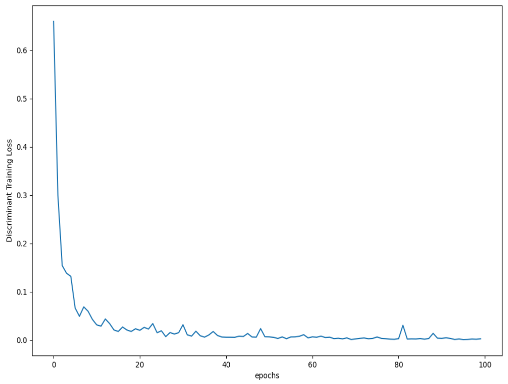


Fig. 20. Supervised Discriminator Training Loss Results

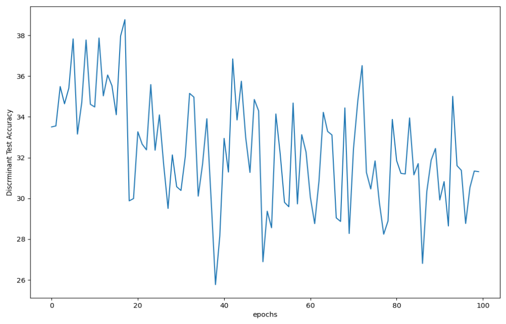


Fig. 21. Supervised Discriminator Testing Accuracy Results

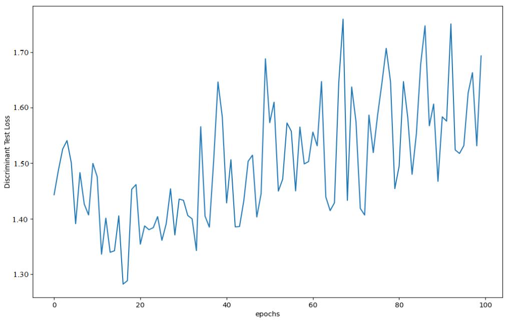


Fig. 22. Supervised Discriminator Testing Loss Results

as there are no pretrained weights and the generator is trained from scratch, per first reason that it is unable to extract features which it in turn it is not able to generate proper fake images thus resulting in low accuracy and high loss of the model.

VI. Conclusion

A. Summary

In this study, we experimented with the ViT, two distinct CNN architectures, and a GAN structure on a feature engineered FER dataset to classify three strong emotions. We may deduce from the ViT that even with increasing the complexity of this structure, good results are difficult to acquire, even after adjusting the hyperparameters. The highest accuracy we achieved using this structure is 55.91%, with a F1 score of 0.55. If we are training it from scratch, we can resolve this by using a dataset better image quality along with a high number of images. Also, using pretrained weights from larger datasets can be used to improve the accuracy. In terms of ST-CNN, we discovered that increasing the batch size improves accuracy. In terms of ResNet-50, we acquired an accuracy of 67.75% when executing the trials, which is lower than the ST-CNN. However, the resulting loss is better than any of the implemented models. In case of SGAN considering the dropout value as 0.4, the Loss function for discriminator as “Sparse_categorical_cross -entropy” has resulted in the accuracy of 38% which is still low.

Based on the data, we discovered that ST-CNN architecture received the best performance with 83.9% for accuracy, 0.68 for loss, and 0.93 for F1 score on the FER dataset.

B. Future research and open problems

FER still faces a challenge in terms of accuracy and real-time detection. For future work, we plan to incorporate the spatial information in ViT with transfer learning. And try out other ensemble architectures. Also, as our dataset is heavily imbalanced, we want to incorporate GAN to generate images that can be used to increase the size of our dataset. Additionally, we want to make the classifier real-time with a high frame per second and high F1 score.

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

Aldahdouh, Alaa. (2020). Emotions Among Students Engaging in Connectivist Learning Experiences. 21. 98-117. 10.19173/irrodl.v21i2.4586.

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

Devlin, J., Chang, M. W., Lee, K., Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding.arXiv preprint arXiv:1810.04805.

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

Hendrycks, D., Gimpel, K. (2016). Gaussian error linear units (gelus).arXiv preprint arXiv:1606.08415.

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

Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., Chen, X. (2016). Improved techniques for training gans.Advances in neural information processing systems,29.

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.

Watson, D., Clark, L. A., Tellegen, A. (1988). Development and validation of brief measures of positive and negative affect: the PANAS scales. Journal of personality and social psychology, 54(6), 1063. Zhong, Y., Deng, W. (2021, April 13). Face transformer for recognition. arXiv.org. Retrieved April 26, 2022 from https://arxiv.org/abs/2103.14803