Facial Emotion Recognition

Faraaz Ahmed
014351779

Manish Arigala
014492712

Tej Chaugule
013856089

Abstract: Facial expression recognition is a topic of great interest in most fields from artificial intelligence and gaming to marketing and healthcare. The goal of this paper is to classify images of human faces into one of seven basic emotions. The problem is inspired from the Facial Emotion Recognition challenge in Kaggle. We approached the solution with a number of different Machine Learning and Deep Learning models including Support Vector Machines (SVM), Decision Trees (DT), k-Nearest Neighbors (kNN), Logistic Regression (LR), Recurrent Neural Networks (RNN) and Convolutional Neural Network (CNN) model. CNNs work better for image recognition tasks since they are able to capture spacial features of the inputs due to their large number of filters. The proposed CNN model consists of six convolutional layers. We introduced data augmentation to our approach whereby it increased the performance of the model, making it the best performing model compared to other models built by other algorithms mentioned previously.

1. Introduction

Facial expressions are manifestations of nonverbal communication. Human beings communicate in the form of speech, gestures, and emotions. We see that the importance of emotion detection lies in various fields like business promotions and security. Business thrives on customer responses to all their products and offers. If an artificial intelligent system can capture and identify real time emotions based on user image or video, they can make decisions based on whether the customer liked or disliked the product or offer. People who use social media to share their experiences or express themselves, primarily use pictures and videos. It can also be seen as a next step of face detection where we may be required to set up a second layer of security, where with face, emotions are also detected. This is useful to verify that the authentic person standing in front of the camera is not just a 2-D representation.

Human emotions are classified as fear, disgust, contempt, surprise, anger, sad, happy, and neutral. These emotions are very elusive. Facial muscle contortions are very minimal to detect the differences and can be very challenging as even a small difference results in different expressions.

The dataset is the FER2013 dataset obtained from Kaggle. The dataset consists of 48x48 grayscale images of faces. All the faces are centrally aligned and occupy around the same amount of space. The objective of this project is to classify human faces into one of the seven emotions. With this in mind, a number of different models were experimented with to get comparable results of the highest accuracy that models can reach.

2. Background

2.1 Decision Tree (DT)

Decision trees are a supervised learning technique that predicts a value based on ”learning” rules based on a set of training data given a set of inputs. It is an enormous tree of if-then-else rules. The decision-making cycle starts at the tree's root and descends by answering to series of yes-no questions. It arrives at a single predicted label at the end of the if-then-else chain. This is the output of a decision tree.

2.2 Convolutional Neural Network (CNN)

A Convolutional neural network is a neural network composed of convolution layers which do the computation by performing convolution. Convolution is a mathematical operation on two functions resulting in a third function. The image is represented as pixel value numbers. The convolution operation takes place on these numbers. We make use of both fully-connected layers as well as convolutional layers. In a fully-connected layer, every node is connected to every other neuron in the network. They are the layers used in standard feed-forward neural networks. In convolutional layers, the connections are made across localized regions. Convolution is effective in classification and image recognition compared to a feed-forward neural network. This is because convolution allows us to take advantage of spatial locality by reducing the number of parameters in a network. Furthermore, convolutional neural networks introduce the concept of pooling that reduces the number of parameters by downsampling. Applications of Convolutional neural networks include robotics, self-driving cars and image recognition.

2.3 Support Vector Machines (SVM)

SVM is one of the most powerful classification algorithms. The idea is to divide the two classes accurately to find an optimal hyperplane. There is also a concept of margin, which is the maximum from both classes avoiding any overlapping between two the class. To achieve better classification results, data that is not linearly separable is mapped into a higher dimension. Kernel functions such as radial basis function (rbf) and polynomial are used for non-linear data. In the case of emotion detection, usually a multi-class SVM is used instead of a binary to detect the emotion expressions. K-fold cross-validation is used to compare different machine learning algorithms and to remove variances in the database. In k-fold cross-validation, the dataset is divided k times into k slices, and prediction results are averaged over all iterations. Principal component analysis (PCA) is used for feature set reduction and then the reduced feature set is fed to SVM. The image feature space is transformed to eigenspace using an eigen matrix in the PCA algorithm. Along with kernel specification, SVM has methods for tuning parameters like C and γ. Here, C is the penalty function for misclassification and gamma helps to optimize the decision boundary. Both these parameters affect the accuracy of the classifiers and can be tuned to get optimal results in both binary and multi-class classification.

2.4 Recurrent Neural Networks (RNN)

Recurrent Neural Network is a generalization of a feed-forward neural network that has an internal memory. RNN is recurrent in nature as it performs the same function for every input of data while the output of the current input depends on the past one computation. After producing the output, it is copied

and sent back into the recurrent network. For making a decision, it considers the current input and the output that it has learned from the previous input. Recurrent Neural Networks can also be divided into units called Long Short Term Memory (LTSM) if there are feedback loops present, or delays of time. The vanishing gradient problem of RNN is resolved here. It trains the model by using back-propagation. RNN can model a sequence of data so that each sample can be assumed to be dependent on previous ones. Recurrent neural networks are even used with convolutional layers to extend the effective pixel neighborhood.

2.5 K Nearest Neighbors (kNN)

Traditional kNN classification algorithms decide about the class of an image by searching for the k images of the training set most similar to the image to be classified, and by performing a class weighted frequency analysis. The k closest images are identified relying upon a similarity measure between images. The kNN based classification methods are modified by relying on images represented by means of local features generated over interest points, as for instance SIFT. With the use of local features and interest points, kNN classification algorithms consider the similarity between local features of the images in the training set rather than the similarity between images, opening up new opportunities to investigate more efficient and effective strategies.

2.6 Logistic Regression (LR)

Logistic Regression is a Supervised Classification Algorithm. Logistic Regression is a regression model that interpolates the dataset using a sigmoidal function where the predicted value is between 0 and 1. In logistic regression, precision is used to predict the next weight values and decision threshold is used. The Activation function of logistic regression converts linear regression equation to a logistic equation. Multiclass Logistic Regression model uses softmax function. Because of its simplicity, logistic regression is a good baseline to compare with other classification Algorithms.

3. Methodology

The task of facial emotion recognition can be broadly classified into two approaches. In the first approach features are extracted from the images. Now the images are represented using the features instead of pixels. After feature extraction, machine learning algorithms such as SVM, KNN, Decision tree are used to achieve the task of emotion recognition. This approach is followed by authors of [1]. The second approach would be to use neural networks to achieve the task of emotion recognition. During recent years, deep learning has made great strides in the field of computer vision. The authors of [2] have followed this approach. We will be following both the approaches. We will apply deep learning-based algorithms such as CNN (Convoluted Neural Networks) and RNN (Recurrent Neural Network) to find the solution to our problem along with the Machine Learning algorithms, SVM (Support Vector Machine), KNN (k Nearest Neighbors), and DT (Decision Tree). The following are the two approaches we took

3.1 Approach 1

- *Preprocessing*:
    - The dataset was randomly split into train and test datasets using sklearn's train_test_split function
- In this approach, we started off with the Scale-Invariant Feature Transform algorithm or simply SIFT. The SIFT algorithm is used to extract key points or features of images that are scale and orientation invariant. We extracted the SIFT features from all the images.
- In the next step, we used the bag of visual words methodology to generate feature vectors for each image in the dataset. The bag of visual words algorithm generates feature vectors based on key points and descriptors. All the feature vectors are clustered to form the visual words/codebook. Each image is represented as a histogram of features.
- Bag of visual words can be extended by constructing a Vector of Locally Aggregated Descriptors(VLAD). VLAD is constructed as follows: 128D SIFT vector is constructed for each image. Each descriptor is then assigned to the closest cluster of a vocabulary of size k. For each of the k clusters, the vector differences between descriptors and cluster centers are accumulated, and the k 128-D sums of residuals are concatenated into a single k × 128 dimensional descriptor. These descriptors are L2 normalised.
- We then trained and built multiple separate models using KNN, SVM, and Decision Trees.
- In the final step, we evaluated every model on the respective test datasets.

3.2 Approach 2

- *Preprocessing:*
    - Min-Max Normalization to reduce dimensions of feature vectors by dividing every pixel value with 255
    - The dataset was randomly split into train and test datasets using sklearn's train_test_split function
- Data augmentation is used to increase the size of training data.
- CNN, RNN, Logistic Regression models are built.
- The models are trained on the respective training dataset and validated on respective validation datasets.
- The models are evaluated using respective test datasets.

4. Analysis

After the dataset was loaded, some of the images of faces from the dataset were visualised as shown below in the figure

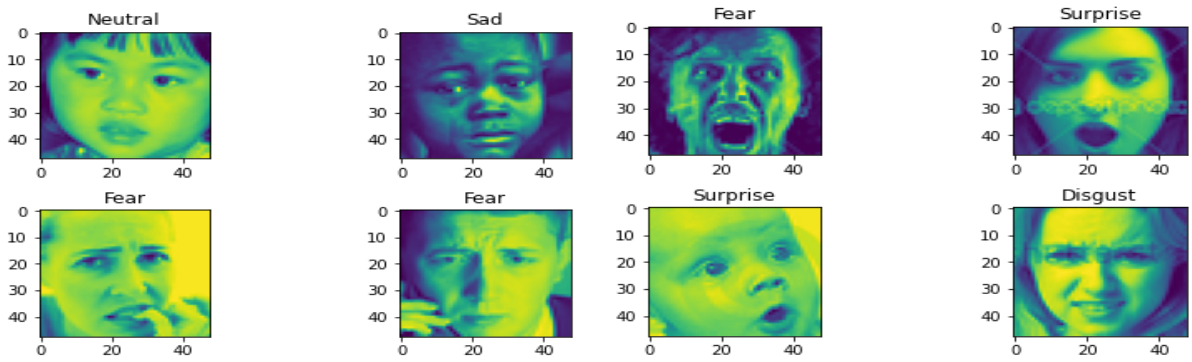


Fig.1 Sample Images from the dataset

The distribution of the dataset is shown below

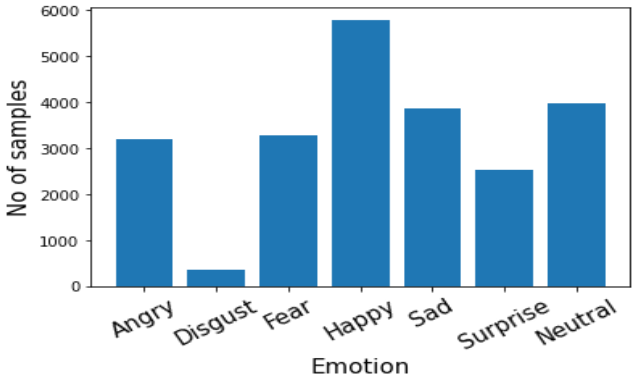
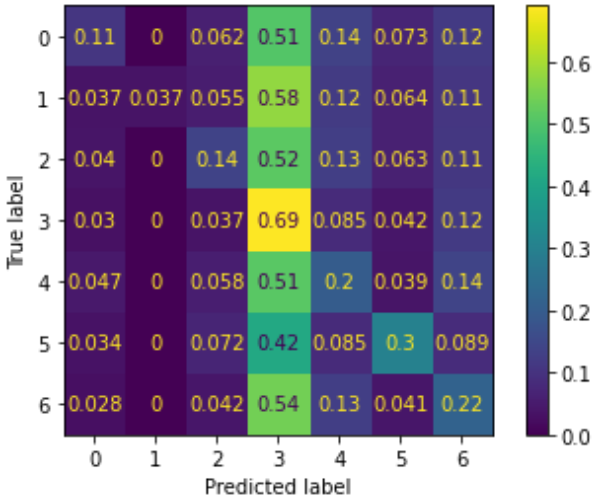


Fig. 2 Dataset Distribution

As seen in the above figure the number of samples belonging to the class disgust are very small. This gives rise to class imbalance problems.

After building, training and testing the models, the following confusion plots were obtained

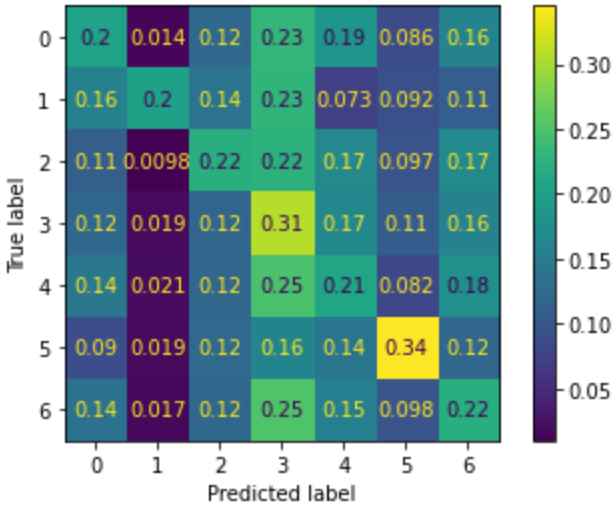
A. Support Vector Machine (SVM) Model:



0-angry; 1-disgust; 2-fear; 3-happy; 4-sad; 5-surprise; 6-neutral

Fig. 3 Confusion matrix plot for SVM

B. Decision Tree (DT) Model

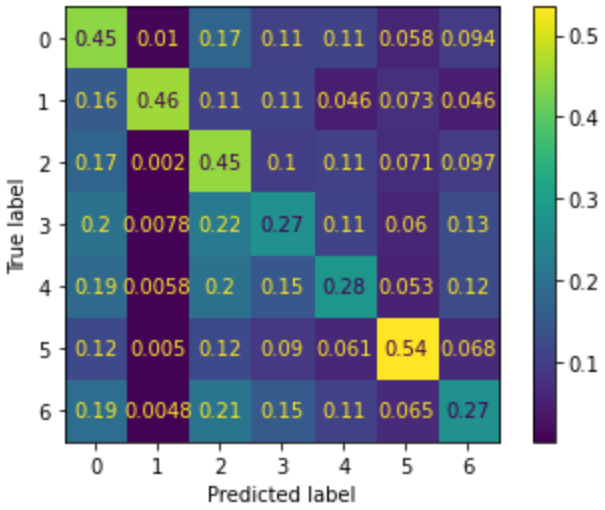


0-angry; 1-disgust; 2-fear; 3-happy; 4-sad; 5-surprise; 6-neutral

Fig. 4 Confusion matrix plot for DT

It can be noted from the above confusion matrices that the traditional machine learning algorithms tend to predict most of the samples to class 3(“Happy” emotion). This cause of this can be related to class imbalance problems as the number of samples belonging to class 3 are very high compared to the other classes.

C. K- Nearest Neighbours

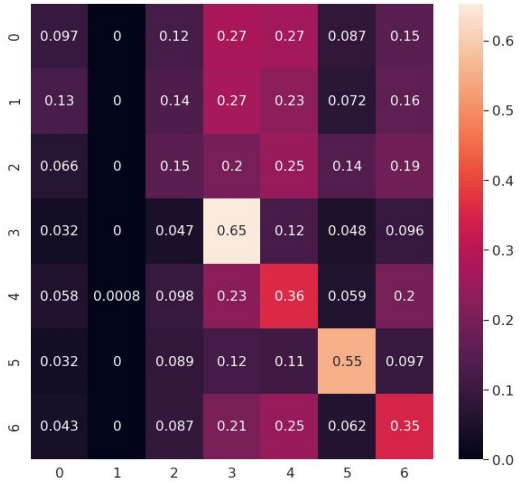


0-angry; 1-disgust; 2-fear; 3-happy; 4-sad; 5-surprise; 6-neutral

Fig. 5 Confusion matrix plot for k-NN

The interesting thing about using VLAD is that all the classes with a smaller fraction of the dataset have a better true positive rate, while the classes with larger fraction of the dataset have lower true positive rate.

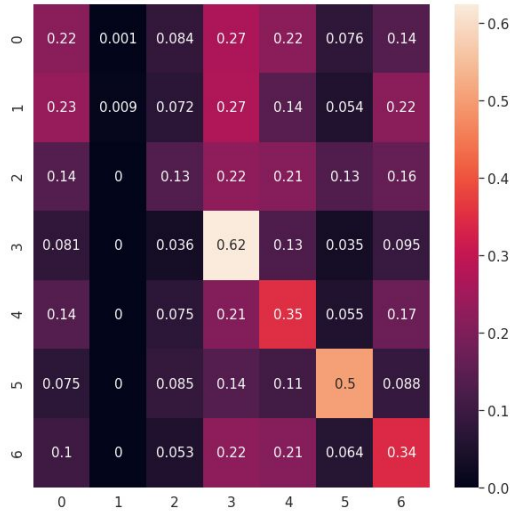
D. Recurrent Neural Networks (RNN) Model



0-angry; 1-disgust; 2-fear; 3-happy; 4-sad; 5-surprise; 6-neutral

Fig.6 Confusion matrix plot for RNN

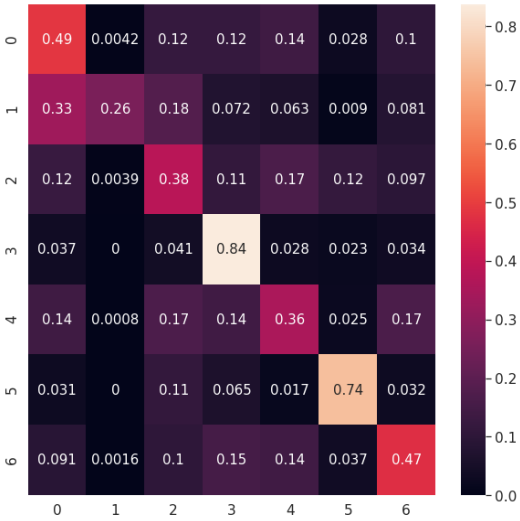
E. Logistic Regression Model



0-angry; 1-disgust; 2-fear; 3-happy; 4-sad; 5-surprise; 6-neutral

Fig.7 Confusion matrix plot for Logistic Regression

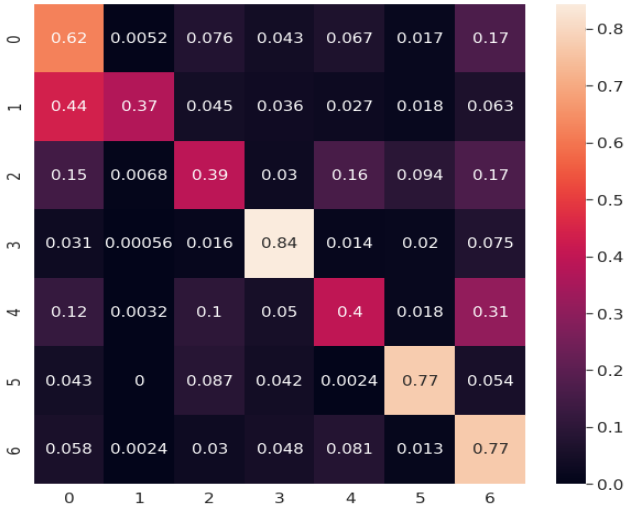
F. CNN Model:



0-angry; 1-disgust; 2-fear; 3-happy; 4-sad; 5-surprise; 6-neutral

Fig.8 Confusion matrix plot for CNN

G. CNN Model with Data Augmentation:



0-angry; 1-disgust; 2-fear; 3-happy; 4-sad; 5-surprise; 6-neutral

Fig. 9 Confusion matrix plot for CNN with Data Augmentation

From the above confusion matrices for CNN we can notice that many samples belonging to class Disgust are predicted as Angry. This can be attributed to high similarity between the two classes as show in the figure below

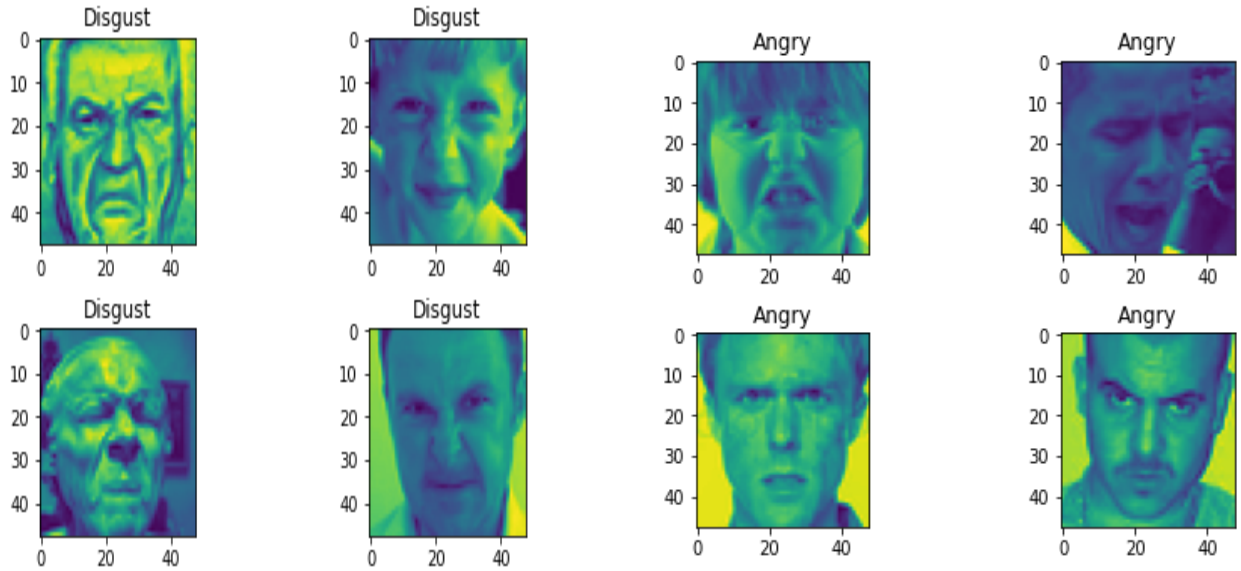


Fig. 10 Similarity between Disgust and Angry Images

Data Augmentation:

Data augmentation is the process of artificially increasing the size of the dataset. Training deep learning neural network models on more data can result in better accuracy. Data augmentation can create variations of the images that can improve the ability to fit

models to generalize well to new images. The ImageDataGenerator method provided by Keras is used for data augmentation. Sample images generated are shown below.

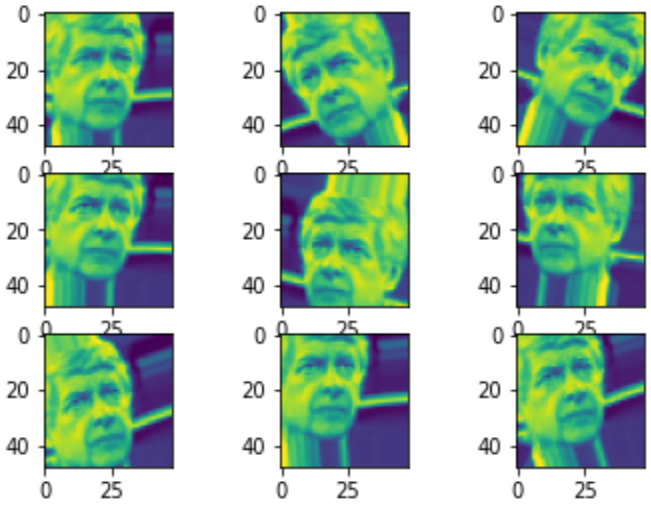


Fig. 11 Example of Data Augmentation

We built the following model which has 11 hidden layers. The model took 125 mins to train. It achieved an accuracy of 64.35% on the test data. The model architecture is shown below

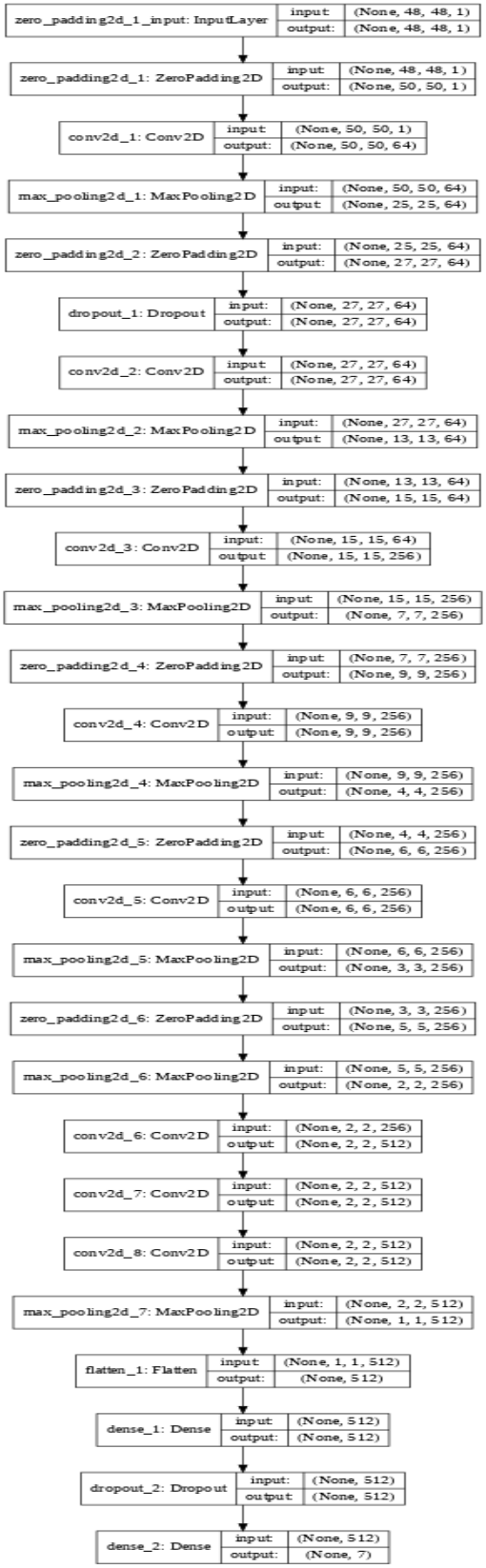


Fig. 12 CNN Model

5. Comparisons

The performance of all the models we built were compared in the following table.

Number	Algorithm	Accuracy
1	Convolution Neural Networks(Data augmentation)	64.35%
2	Convolution Neural Networks	55.84 %
3	Logistic Regression	38.49%
4	Recurrent Neural Networks	38.35%
5	KNN	35.52%
6	Support Vector Machine	33.18%
7	Decision Tree	25.09%

Fig. 13 Comparison Table

As we can see from the above table, CNN based models performed overwhelmingly better than all the other models. Furthermore, data augmentation effectively increased the performance by increasing the accuracy score from 55.84 % to 64.35%.

Comparing the accuracy scores from the Kaggle leaderboard, our best performing model, CNN (Data Augmentation) algorithm would be placed in the top 10 rankings.

6. Conclusion and Future Work

Although we were successfully able to build different Machine Learning models to solve the Facial Emotion Recognition problem, traditional Machine Learning algorithms failed to perform well on the dataset. CNN outperformed other machine learning algorithms. Data augmentation significantly improved the performance of the model. The accuracy can be increased by solving the class imbalance problem.

References

[1] R. T. Ionescu, M. Popescu, and C. Grozea. Local Learning to
Improve Bag of Visual Words Model for Facial Expression Recognition. In *Proceedings of ICML Workshop on Chal- lenges in Representation Learning*, 2013
[2] M. Georgescu, R. T. Ionescu and M. Popescu, "Local Learning With Deep and Handcrafted Features for Facial Expression Recognition," in IEEE Access, vol. 7, pp. 64827-64836, 2019.

[3] Arandjelovic, Relja, and Andrew Zisserman. "All about VLAD." *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*. 2013.