**Facial Emotion Recognition using Deep Learning**

**Uttara Dabbiru,**

**College of Business & Economics**

**California State University East Bay**

**Hayward, California, United States**

[**udabbiru@horizon.csueastbay.edu**](mailto:udabbiru@horizon.csueastbay.edu)

**Anjul Jain,**

**College of Business & Economics**

**California State University East Bay**

**Hayward, California, United States**

**ajain24@horizon.csueastbay.edu**

**Kavya Pothula,**

**College of Business & Economics**

**California State University East Bay**

**Hayward, California, United States**

**kpothula**[**@horizon.csueastbay.edu**](mailto:udabbiru@horizon.csueastbay.edu)

**Manushree Thimmaiah Sannathimmappa,**

**College of Business & Economics**

**California State University East Bay**

**Hayward, California, United States**

**mthimmaishsannathimmappa@horizon.csueastbay.edu**

I. ABSTRACT

**The Facial Expression Recognition (FER) system is the process of identifying the emotional state of a person. In general, facial gestures are normal and clear means of expressing their feelings and intentions to human beings. The main features of non-verbal conversations are facial expressions. Automatic facial expression recognition systems have many applications including, but not limited to, human behavior understanding, and detection of mental disorders. In this project, various deep learning methods have been used to identify the key seven human emotions: anger, disgust, fear, happiness, sadness, surprise, and neutrality.**

**In this project, facial expression recognition systems are implemented using Convolution Neural Network (CNN) and VGG16 models with various capacities and different activation functions. Kaggle facial expression dataset along with a few images taken by us are included to train the models. The training dataset has 28,709, and the testing dataset has 7179 images.**

**The system achieved 76.22% of training accuracy and 56.56% of validation accuracy using the VGG16 model with increased capacity. This is the best accuracy, with lower generalization error, out of all the models.**

***Keywords - Deep Learning(DL), Convolutional Neural Network(CNN), Transfer Learning(TL), Facial Emotion Recognition(FER)***

II. INTRODUCTION

Facial Emotion Recognition is a technology used for analyzing sentiments by different sources, such as pictures and videos. It belongs to the family of technologies often referred to as ‘affective computing’, a multidisciplinary field of research on computer’s capabilities to recognise and interpret human emotions and affective states and it often builds on Artificial Intelligence technologies. [1]

The appreciation of human expression plays an important role in interpersonal relations. Emotions are conveyed by voice, hand and body movements, and facial expressions. Therefore the interaction between human and computer contact is of high importance for extracting and interpreting emotions.[2]

In our project, facial emotion recognition is used for classifying images into one of the 7 categories of emotions. Our problem at hand is to classify the emotions into one of the following: Neutral, Happy, Sad, Surprised, Angry, Fearful or Disgust. The output must be as dynamic as the expression of the face changes. An individual’s intent within a social situation becomes crucial as the systems can adapt their responses and behavioral patterns. According to the emotions of the humans, the intelligent systems can use emotion recognition to improve their interactions and make the interaction more natural.

Potential uses of FER cover a wide range of applications in various fields such as Healthcare, Employment, Education, Public Safety, Crime detection

Coming to the model description, we initially run the model using a Convolution Neural Network (CNN) and then VGG 16.

A Convolutional Neural Network (ConvNet/CNN) is a Deep Learning algorithm which can take in an input image, assign importance (learnable weights and biases) to various aspects/objects in the image and be able to differentiate one from the other. The pre-processing required in a ConvNet is much lower as compared to other classification algorithms. While in primitive methods filters are hand-engineered, with enough training, ConvNets have the ability to learn these filters/characteristics. It is a collection of two types of layers, Feature Extraction part and the Classifier part.



As a next step we plan to use the pretrained VGG16 model by experimenting the hyperparameters by ,

1. Increasing capacity

2. Changing the activation function

The VGG model, or VGGNet, that supports 16 layers is also referred to as VGG16, which is a convolutional neural network model. The VGGNet-16 supports 16 layers and can classify images into 1000 object categories, including keyboard, animals, pencil, mouse, etc. Additionally, the model has an image input size of 224-by-224. This means that VGG16 is a pretty extensive network and has a total of around 138 million parameters.

We plan to experiment with different architectures of CNN and CNN-LSTM proposed by different researchers and present some different databases containing spontaneous images collected from the real world, to have and achieve an accurate detection of human emotions.

Include CNN and VGG brief explanation

CNN

Transfer learning for facial detection.

The emotional state of humans can be obtained from verbal and non-verbal information captured by the various sensors, for example from facial changes, tone of voice and physiological signals. Face changes during a communication are the first signs that transmit the emotional state, which is why most researchers are very interested in this modality. Extracting features from one face to another is a difficult and sensitive task to have a better classification.

III. METHODOLOGY

*A. Data Overview:*

The sample data is made up of images of seven different categories of facial emotion. We randomly split the images for each emotion into training (80%) and validation (20%) sets.

Examples of images of each class can be seen in the image below:



For ease of use in Keras, our folder structure first splits into training and validation, and each of those are split again into directories based upon the seven facial emotion categories. All images are of size 20 x 20 pixels. The training dataset has 6507, and the testing dataset has 1922 images

*B. Data Pipeline:*

Before we get to building the network architecture, & subsequently training & testing it we need to set up a pipeline for our images to flow through, from our local hard-drive where they are located, to, and through our network.

In the code, we:

* Import the required packages
* Set up the parameters for our pipeline
* Set up our image generators to process the images as they come in
* Set up our generator flow - specifying what we want to pass in for each iteration of training

We specify that we will resize the images down to 32 x 32 pixels, and pass in 32 images at a time (known as the batch size) for training. To start with, we simply use the generators to rescale the raw pixel values (ranging between 0 and 255) to float values that exist between 0 and 1. The main reason to do this is to help Gradient Descent find an optimal, or near optional solution each time much more efficiently. In other words, it means that the features that are learned in the depths of the network are of a similar magnitude, and the learning rate that is applied to descend down the loss or cost function across many dimensions, is somewhat proportionally similar across all dimensions. This means. training time is faster as Gradient Descent can converge faster each time.

With this pipeline in place, the images will be extracted in batches of 32 from our hard-drive, where they’re being stored and sent into our model for training.

*C. Network Architecture*

a) CNN - Model 1

Our baseline network is a CNN model which gives us a starting point to refine from. This network contains 4 Convolutional Layers, each with filters ranging from 16 to 128 and subsequent Max Pooling Layers and Dropout at 25% for two layers and at 50% for one layer. We have a single Dense (fully connected) layer followed by flattening with 16 neurons and then followed by an output layer. We apply the relu activation function on all layers, softmax for output layer, and use the adam optimizer with a very slow learning rate of 0.0001.

Training The Network

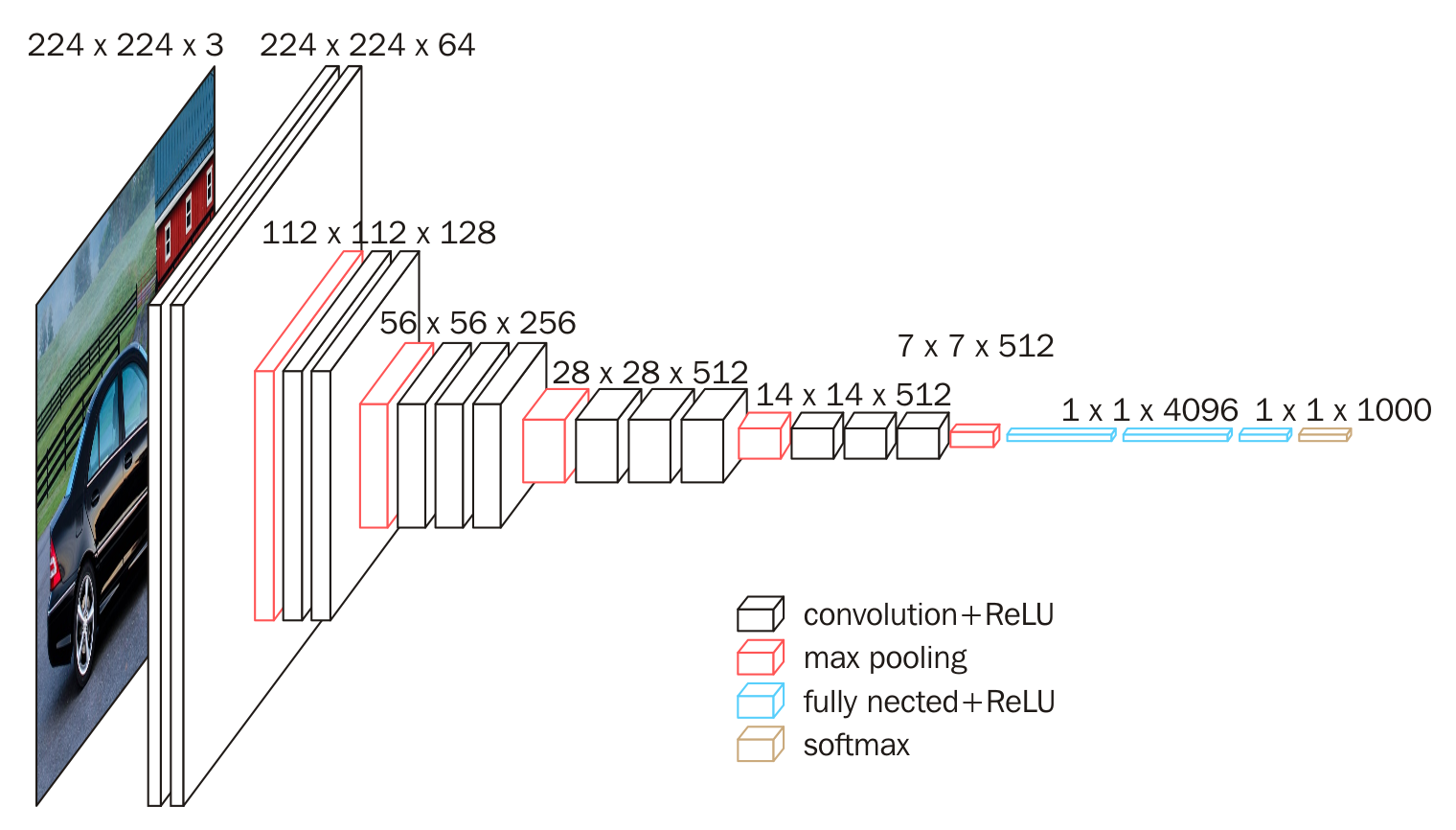
With the pipeline, and architecture in place, we are now ready to train the baseline network. The steps are as follows:

* Specify the number of epochs for training
* Set a location for the trained network to be saved (architecture & parameters)
* Train the network and save the results to an object called history

b) VGG16 - Model 1

Keras makes the use of VGG16 very easy. We will download the bottom of the VGG16 network (everything up to the Dense Layers) and add in what we need to apply the top of the model to our facial emotion classes.

We then need to specify that we do not want the imported layers to be re-trained, we want their parameters values to be frozen.



The original VGG16 network architecture contains two massive Dense Layers near the end, each with 4096 neurons. Since our task of classifying 7 types of emotions is more simplistic than the original 1000 ImageNet classes, we reduce this down and instead implement 5 Dense Layers with neurons ranging from 16 to 128, followed by our output layer.

VGG16 architecture has a total of 14.8 million parameters, much bigger than what we have built so far. Of this, 7.6 million parameters are frozen, and 7.2 million parameters will be updated during each iteration of back-propagation, and these are going to be figuring out exactly how to use those frozen parameters that were learned from the ImageNet dataset, to predict our classes of facial emotions.

c) VGG 16 - Model 2

Here, we have used the VGG-16 network with increased capacity.

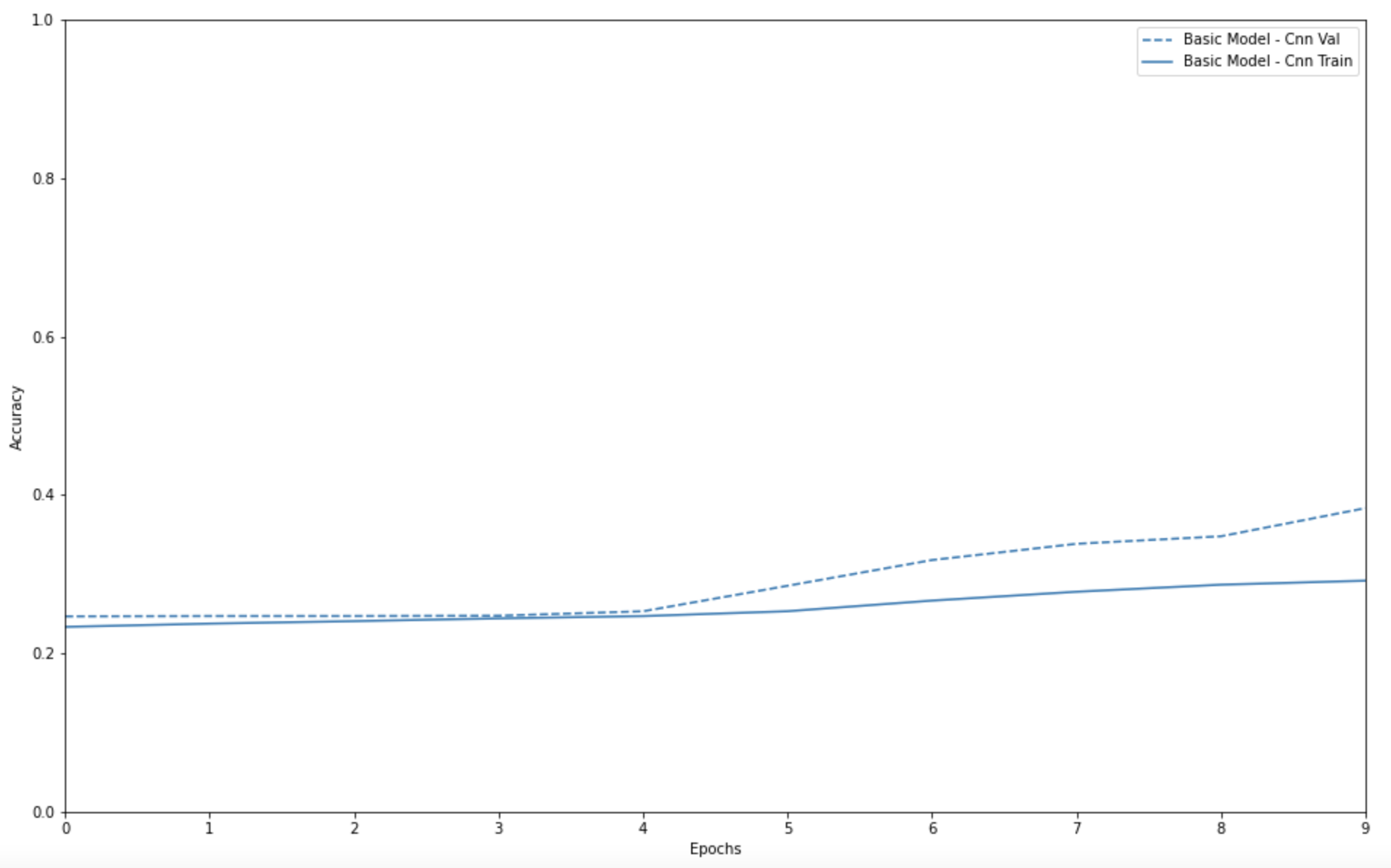
We have experimented with VGG16 architecture that originally had a total of 14.8 million parameters by adding more layers and neurons making the network deeper and wider. With increased capacity, the network has a total of 16.2 million parameters. Of this, 7.6 million parameters are frozen, and 8.6 million parameters will be updated during each iteration of back-propagation, and these are going to be figuring out exactly how to use those frozen parameters that were learned from the ImageNet dataset, to predict our classes of facial emotions. With more parameters to train, the model took longer time to train but gave the best validation accuracy.

IV. RESULTS & DISCUSSION

As we saved our training process to the history object, we can now analyze & plot the performance (Classification Accuracy) of the updated network epoch by epoch.

The below images show the epoch by epoch Classification Accuracy, for both the training set (solid line) and the validation set (dashed line) for all the three models we have trained.

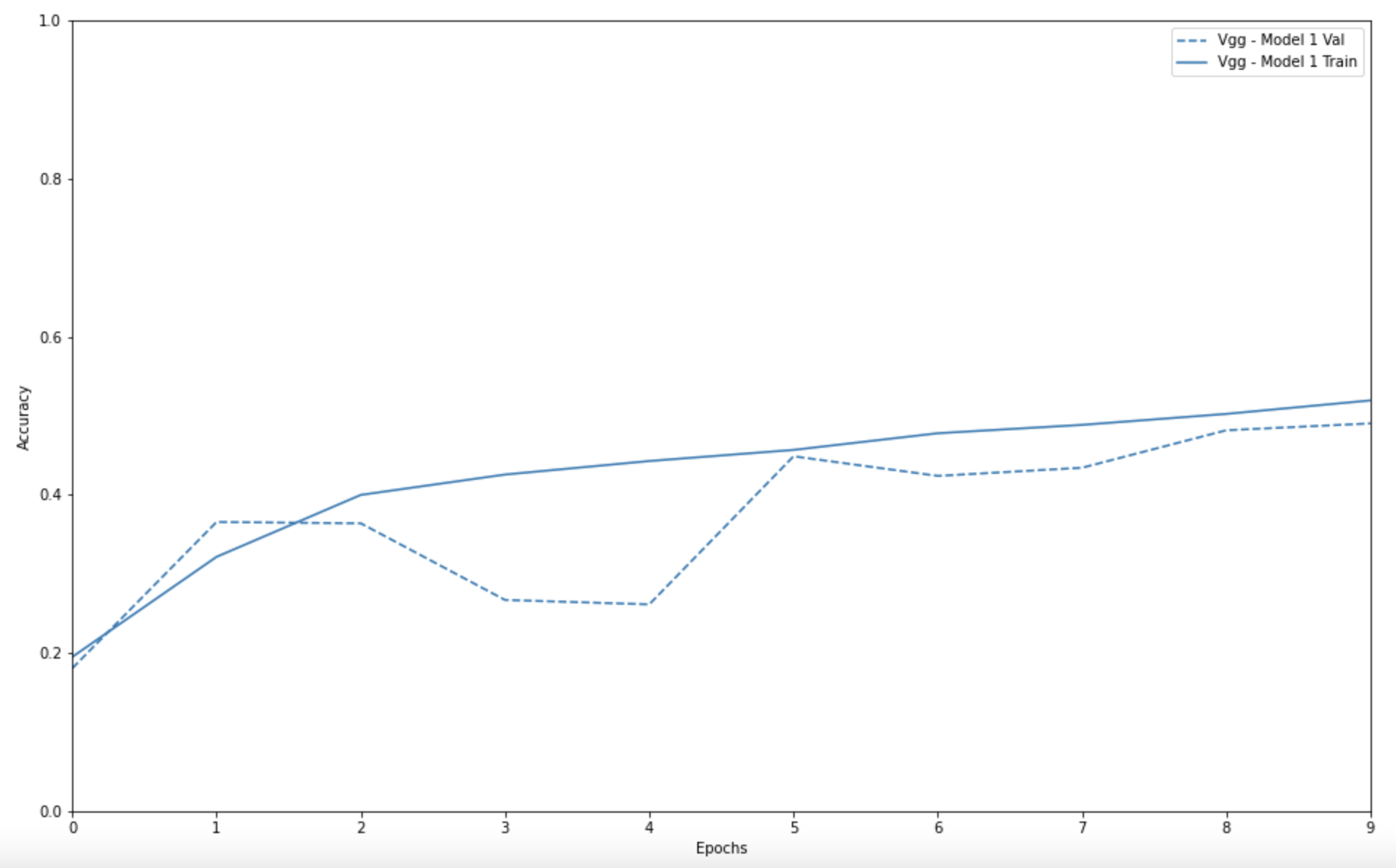
a) CNN Model 1:



According to the above graph, the classification accuracy on the validation set is around 31%. The gap between the classification accuracy on the training set, and the validation set is minimal. The two lines are trending up at the same rate across all epochs of training and we see the generalization that we want.

The addition of dropout has remedied the overfitting. This is because, while some neurons are turned off during each mini-batch iteration of training, just in a way where no neuron, or combination of neurons will become so hard-wired to certain features found in the training data.

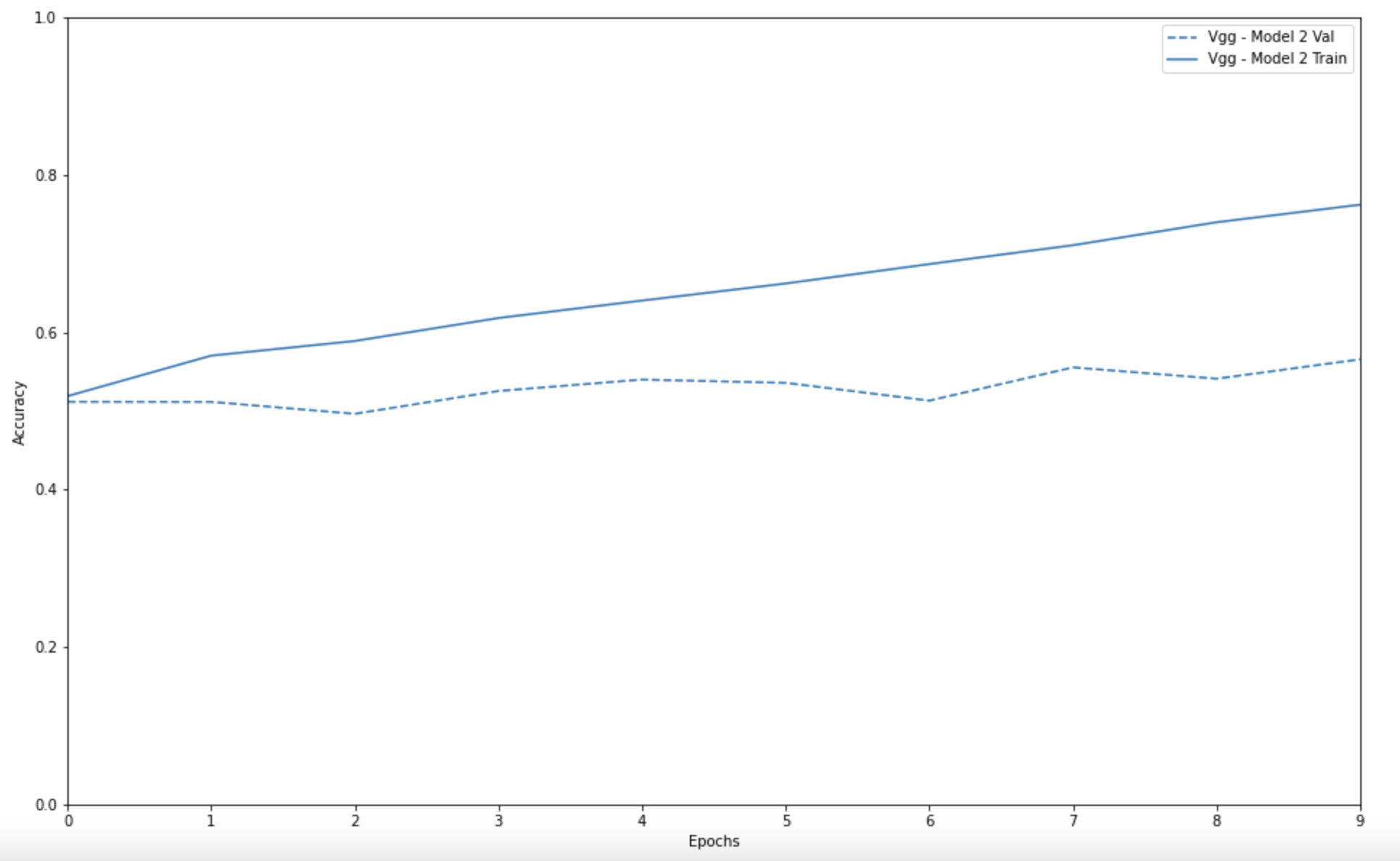
b) VGG-16 Model 1:



As we again saved our training process to the history object, we can now analyze & plot the performance (Classification Accuracy, and Loss) of the updated network epoch by epoch.

During training, we have assessed our performance on both training and validation sets. In the above graph, we observe a peak in classification accuracy on validation sets of around 49% which is a significant improvement from the previous basic CNN model.

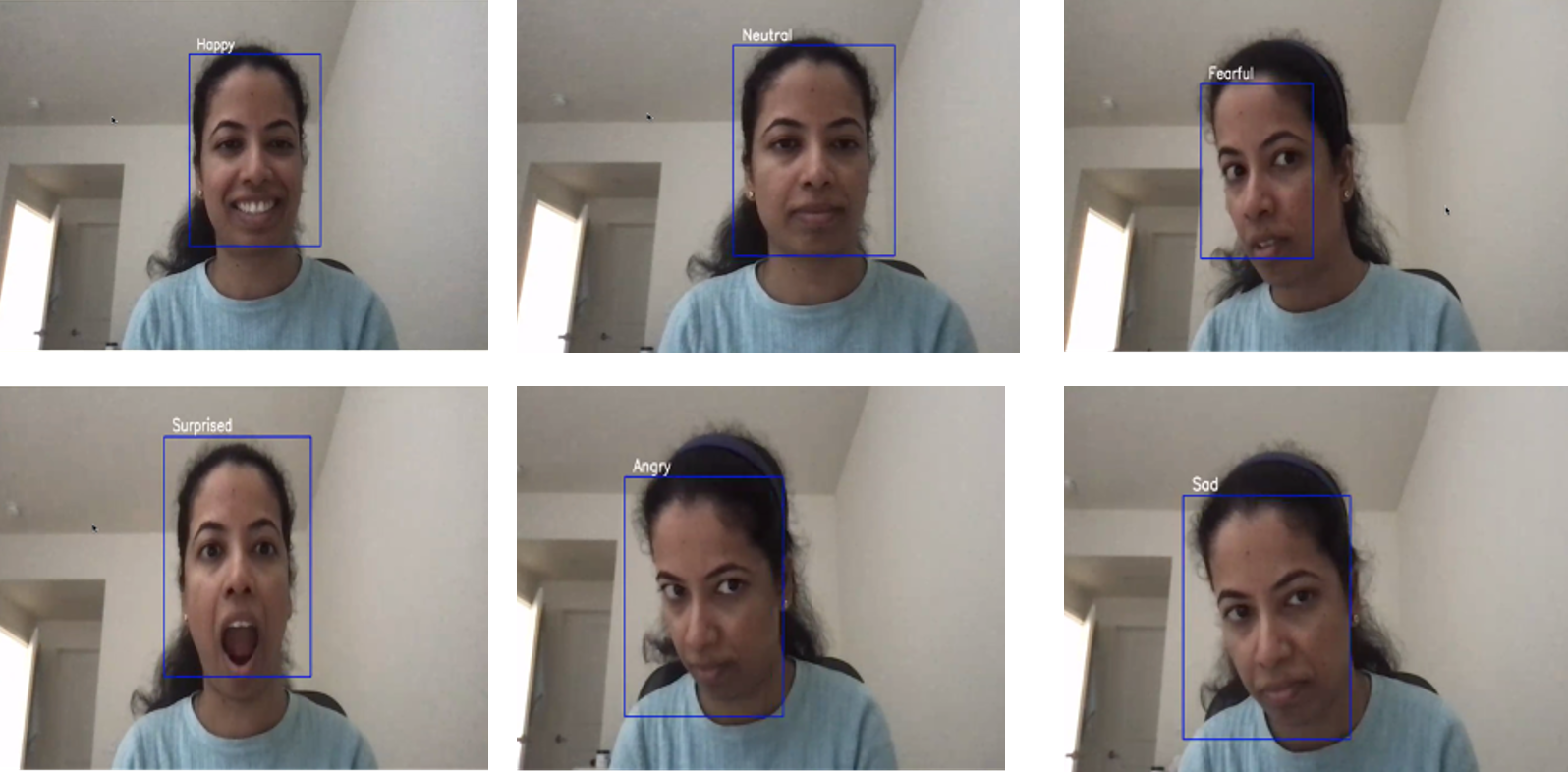
c) VGG-16 Model 2:



In the above graph, we observe that with an increased model capacity, we have got the best classification accuracy on validation set of 57% with VGG-network 2. Adding more parameters would only risk classifying the images incorrectly. The model tends not to generalize the features which might lead to generalization error. This model is able to recognize almost all the features correctly.

Summarized below is a comparison table of training & validation accuracies for the above 3 network:

|  | CNN | VGG-1 | VGG-2 |
| --- | --- | --- | --- |
| Training Accuracy | 29% | 52% | 76% |
| Validation Accuracy | 38% | 49% | 57% |



All emotions except Disgust were accurately recognized. *Neutral, happy* and *surprised* emotions were the most accurate.

In the above image, emotions like *neutral, happy* and *surprised* were the most accurately recognized & *disgust* emotion was least accurately recognized by the VGG-16 model 2.

**V. CONCLUSION**

**VI: REFERENCE**

[1]VEMOU, Konstantina, and Anna HORVATH. EDPS TechDispatch on Facial Emotion Recognition. 2021, https://edps.europa.eu/system/files/2021-05/21-05-26\_techdispatch-facial-emotion-recognition\_ref\_en.pdf.