Capstone Project

Live Class Monitoring System
(Face Emotion Recognition)

by

Mahin Arvind Chanthira Sekaran

Content

- Introduction
- Problem Statement
- Dataset Description
- Data Augmentation
- Model Compilation
- Evaluation Metrics
- Results
- Final Report
- Deployment
- Conclusion
- References

Introduction

- Face emotion recognition technology detects emotions and mood patterns invoked in human faces.
- This technology is used as a sentiment analysis tool to identify the six universal expressions, namely, happiness, sadness, anger, surprise, fear and disgust from a neutral face.
- Identifying facial expressions has a wide range of applications in human social interaction detection for industries like digital learning, market research and mental health.
- In this project, the Face Emotion Recognition 2013 Dataset was used to train five different types of architectures built using convolutional layers.
- The best-performing model was deployed as a Streamlit application to perform real-time emotion recognition.

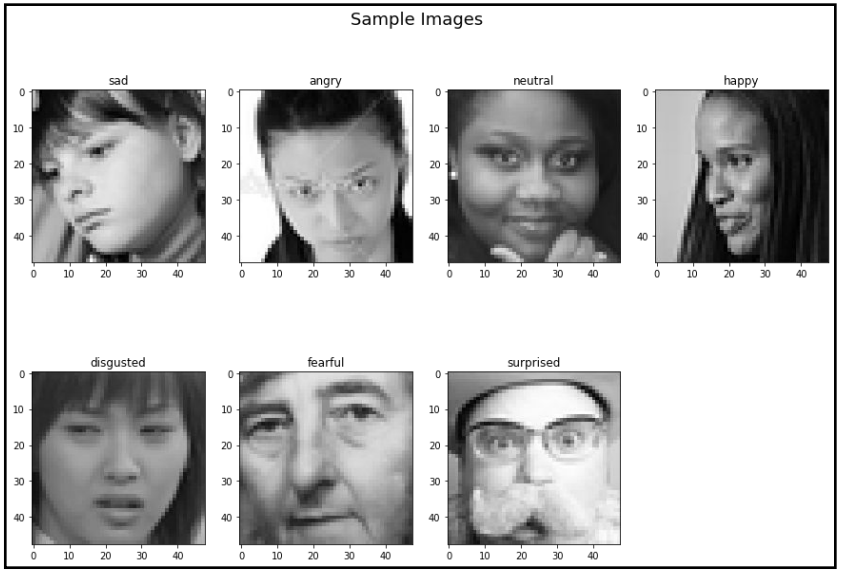
Problem Statement

In this project we'll be using the Face Emotion Recognition 2013 dataset to

1. Effectively identify student emotions using minimum reference images.
2. Perform face emotion recognition on a webcam video feed in real-time.
3. Create a web application to access the model.
4. Deploy model on the cloud as an end-to-end solution.

Data Description

- The FER-2013 dataset was created by gathering the results of a Google image search of each emotion.
- This dataset consists of 22,965 labelled images in the training set, and 5,741 labelled images in the validation set while 7,178 images were used as the test set for evaluation.
- Each image in FER-2013 is labelled as one of seven emotions: happy, sad, angry, afraid, surprise, disgust, and neutral.
- The images in FER-2013 consist of both posed and unposed headshots in grayscale and are 48 x 48 pixels in dimension.



Data Augmentation

- Image data augmentation is a technique that can be used to artificially expand the size of a training dataset by creating modified versions of images.
- Augmentation techniques can create variations of the images that can improve the ability of the models to generalize what they have learned towards new images.
- Zoom, Horizontal Orientation and Brightness were the randomly augmented image features
- Apart from augmenting images, all the image pixel values in the dataset are rescaled from the [0, 255] range to the [0,1] range.

Feature	Range
Zoom	[0, 0.2]
Horizontal Flip	True/False.
Brightness	[0.2, 1.2.]

Evaluation Metrics

Categorical Cross entropy is the sum of the product of the predicted value and the logarithm of the target value.

Categorical Accuracy calculates the percentage of predicted values that match with actual values for one-hot labels.

Confusion Matrix is the table that contains the list of true positives, true negatives, false positives and false negative occurrences

Precision explains how many of the correctly predicted cases actually turned out to be Positive

Recall explains how many of the actual positive cases we were able to predict correctly with our model.

F1 score combines the precision and recall of a classifier into a single metric by taking their harmonic mean.

Model Compilation

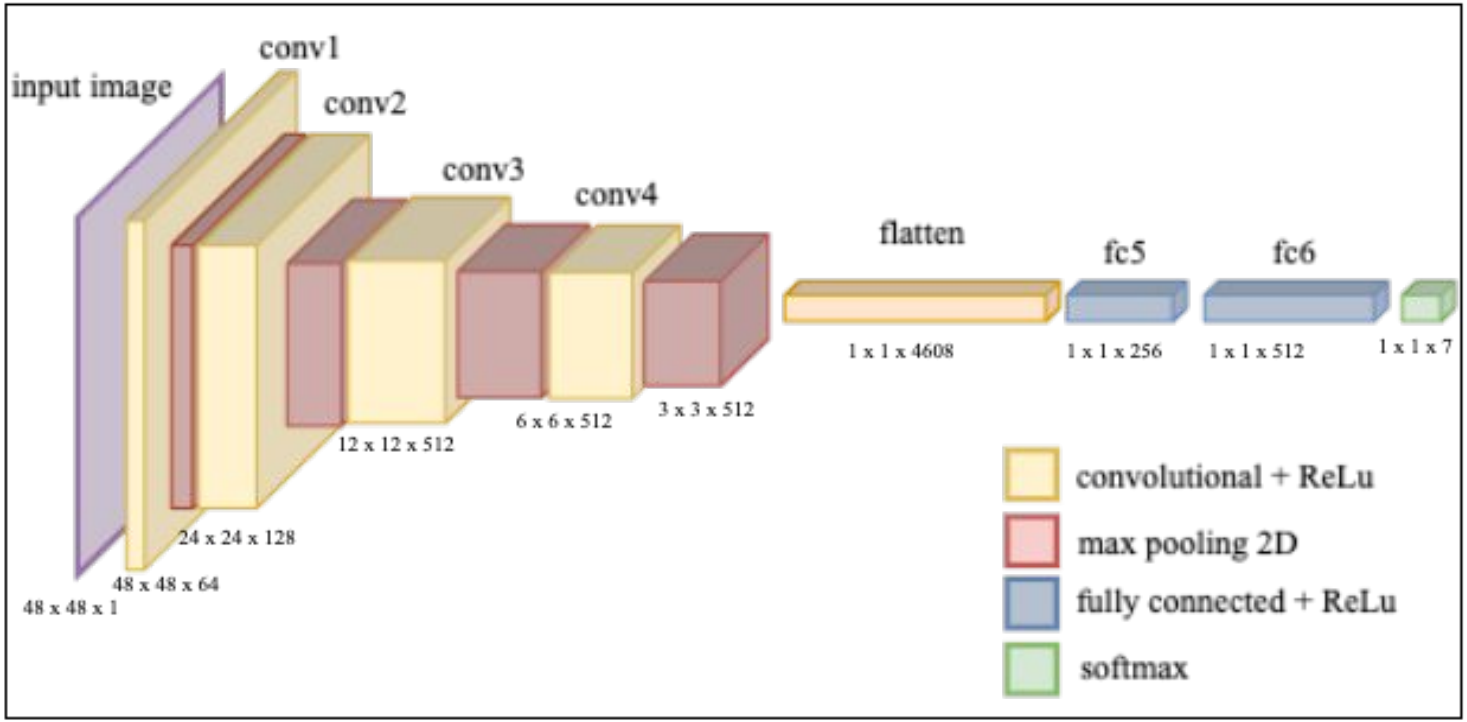
- During the model compilation and training phase, the loss and metrics monitored were categorical cross entropy and categorical accuracy.
- Adam is a replacement optimization algorithm for stochastic gradient descent for training deep learning models. It combines the properties of the AdaGrad and RMSProp. Adam Optimization was used with a learning rate of 0.001.
- The model was trained in data with a batch size of 64 for a period of 50 epochs.
- For efficient training, early stopping was used to stop training when the delta value of monitored validation loss was less than zero for ten consecutive epochs.
- A callback for reducing the learning rate by a factor of 0.02 was used to find global minima when the validation loss was observed to plateau.

Observations

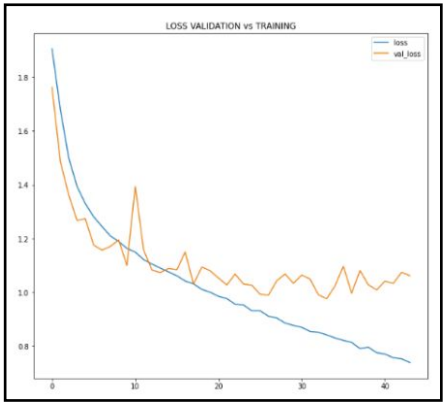
Generic ConvNet

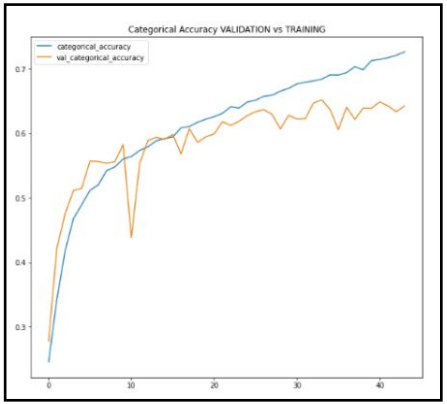
- The generic CNN was arbitrarily designed with 4 convolutional layers activated using the ReLu function.
- Convolutional filters of sizes 3x3 and 5x5 were used. Each level used batch normalization and a max-pooling layer for efficient computation.
- After flattening, two dense layers and a softmax layer were used to classify the image into one of the seven universal facial expressions.
- As a regularisation measure, a 25 % dropout was enforced after each ReLu activation function.
- The Generic ConvNet achieved a training categorical accuracy of 68.42 % and a validation categorical accuracy of 65.2 %. When evaluated with the test dataset, the model produced a weighted F1 score of 64.38 % and a balanced accuracy score of 61.12 %.

Generic ConvNet Architecture



Generic ConvNet Architecture



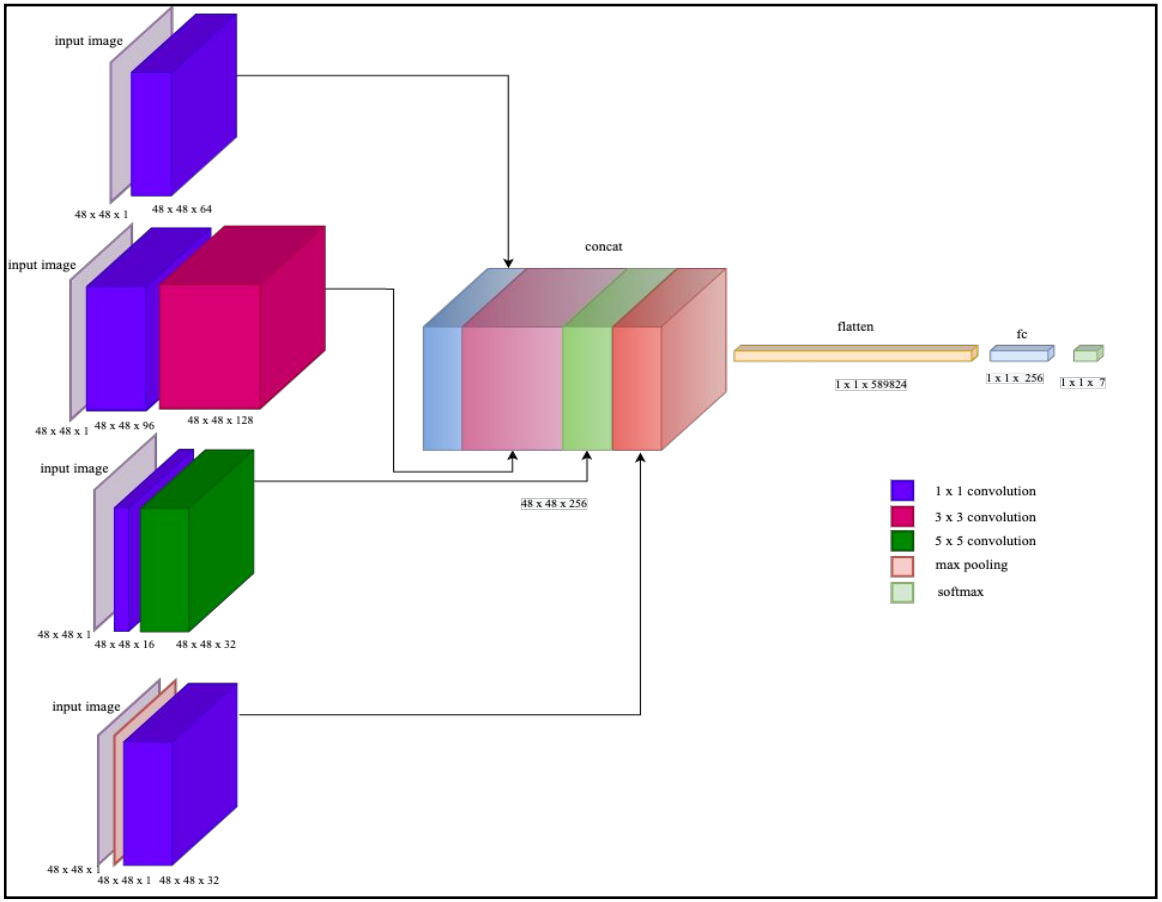


Conv Model	Recall	Precision	F1 score	Balanced Accuracy
Generic	0.6473	0.6493	0.6438	0.6118

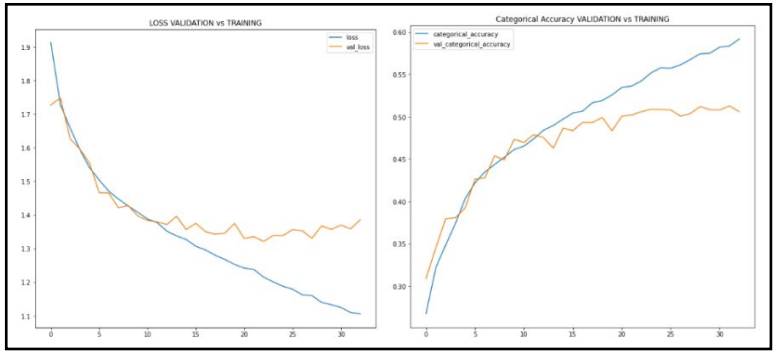
Inception block ConvNet

- The inception module was described and used in the GoogLeNet model.
- The key innovation of the inception model is the use of a block of parallel convolutional layers with different sized filters (1×1, 3×3, 5×5) and 3×3 max pooling layer. The results of the layers are concatenated.
- This is a simple and powerful architectural unit that allows the model to learn parallel filters of the same size and differing sizes, allowing learning at multiple scales.
- After flattening the inception block output, a dense layer and a softmax layer are implemented.
- The Inception block ConvNet achieved a training categorical accuracy of 58.36 % and a validation categorical accuracy of 51.3 % before plateauing . When evaluated using the test dataset, the model produced a weighted F1 score of 49.94 % and a balanced accuracy score of 46.62 %.

Inception block ConvNet Architecture



Inception block ConvNet Architecture



Conv Model	Recall	Precision	F1 score	Balanced Accuracy
INCEPTION Block	0.5139	0.5004	0.4994	0.4662

VGG block ConvNet

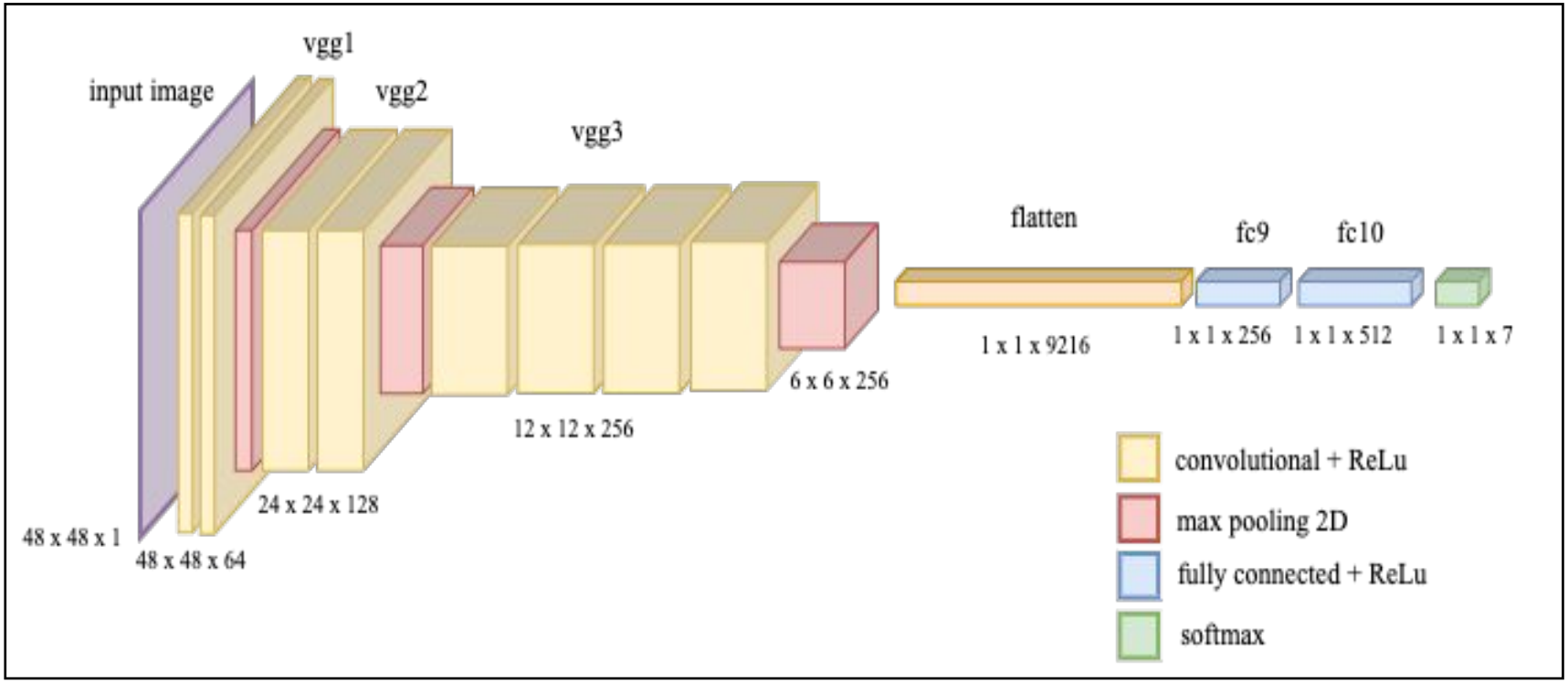
- The key innovation in this architecture was the definition and repetition of VGG-blocks. These are groups of convolutional layers that use small filters (3×3 pixels) followed by a max pooling layer.

- In this model, each VGG block consisted of a specified number of convolutional operations producing one among 64, 128 and 256 output filters. These convolutional operations used three-by-three filters and performed the same convolution. The combination of convolutional layers was followed by a max-pooling with two-by-two filters and with the same padding.

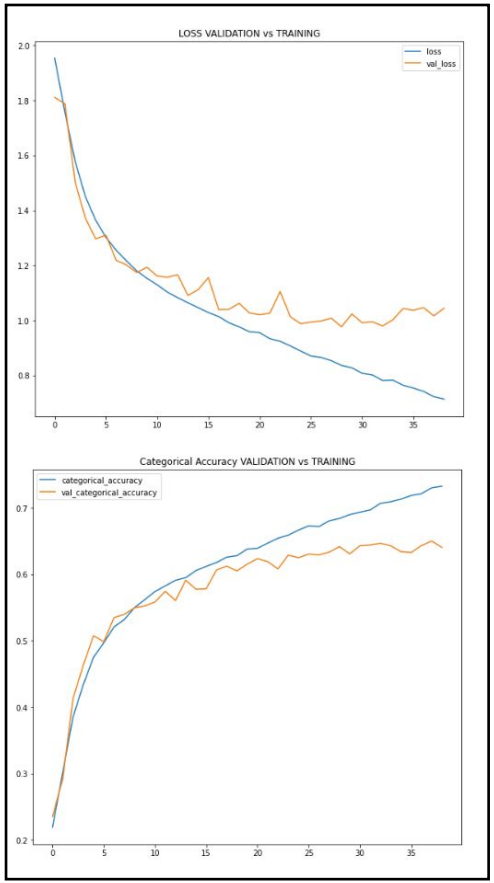
- Each VGG block was followed by a batch normalization layer and a dropout layer for regularisation. After flattening, two dense layers and a softmax layer were used to recognize face emotion.

- The VGG block ConvNet achieved a training categorical accuracy of 72.30 % and a validation categorical accuracy of 65.01 % before plateauing. When evaluated using the test dataset, the model produced a weighted F1 score of 64.95 % and a balanced accuracy score of 62.26 %.

VGG block ConvNet Architecture



VGG block ConvNet Architecture



Conv Model	Recall	Precision	F1 score	Balanced Accuracy
VGG Block	0.6509	0.6527	0.6495	0.6226

VGG-16 with ImageNet weights

- VGG-16 is a convolutional neural network that is 16 layers deep. With the help of transfer learning, it is used to load a pre-trained version of the network trained on more than a million images from the ImageNet database.

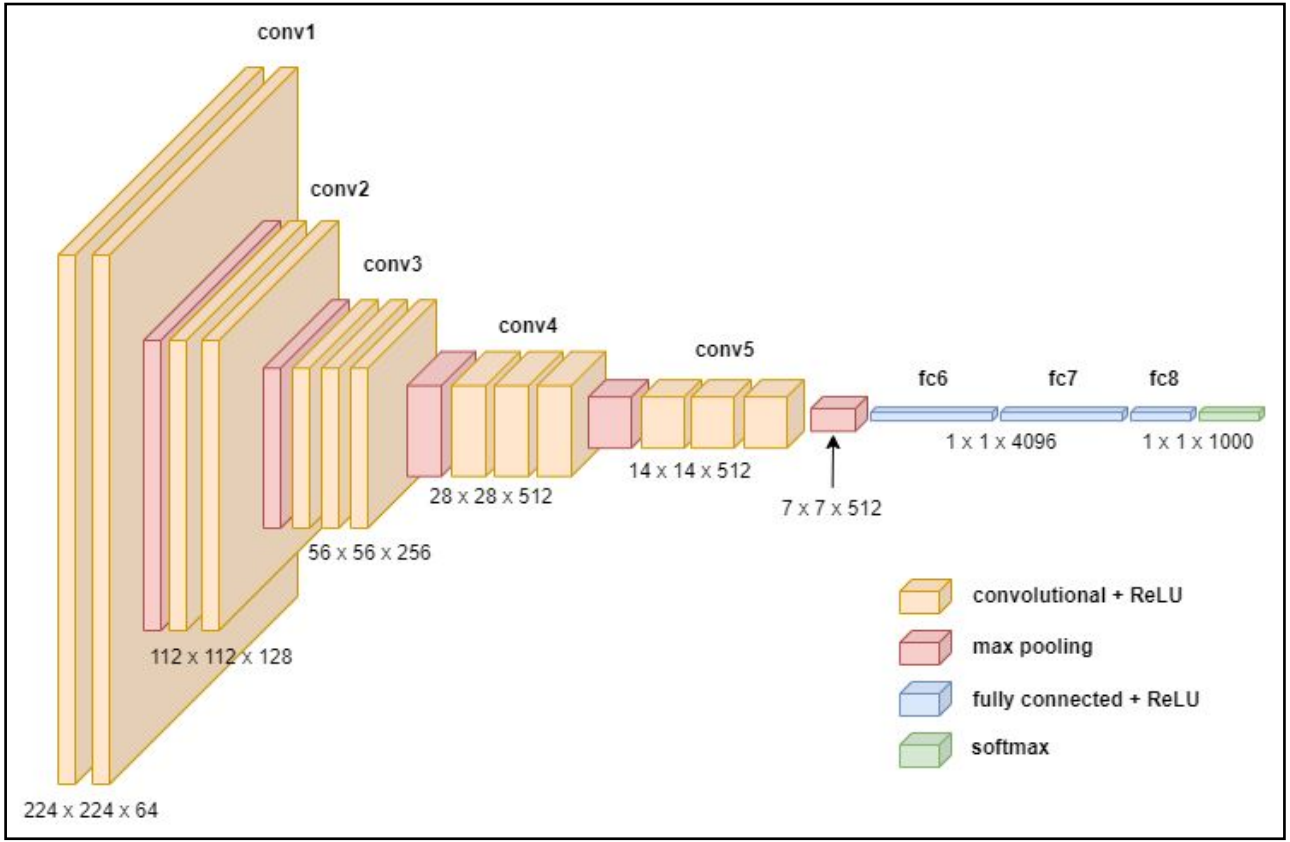
- The pre-trained network can classify images into thousand object categories.

- There are 13 convolutional layers, five Max Pooling layers, and 3 Dense layers which sum up to 21 layers in the VGG-16 model of which there are sixteen learnable layers.

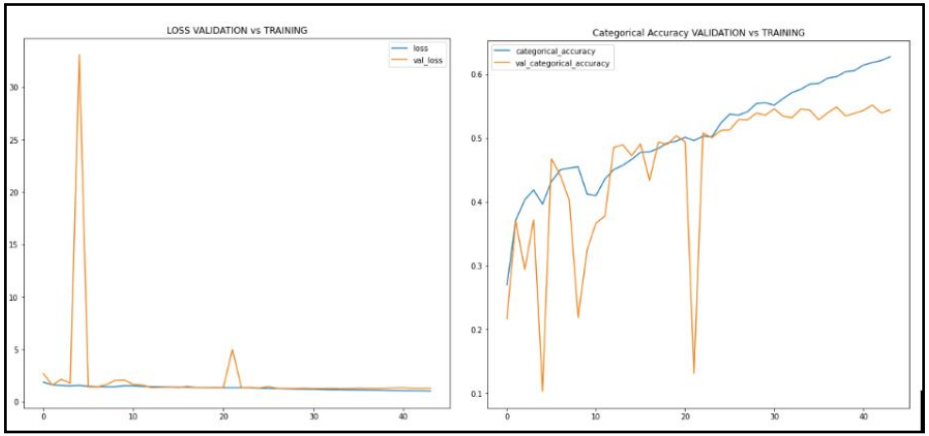
- The VGG-16 architecture with ImageNet weights achieved a training categorical accuracy of 61.81 % and a validation categorical accuracy of 55.14 %.

- When evaluated using the test dataset, the model produced a weighted F1 score of 54.22 % and a balanced accuracy score of 52.47 %.

VGG-16 block ConvNet Architecture



VGG-16 block ConvNet Architecture

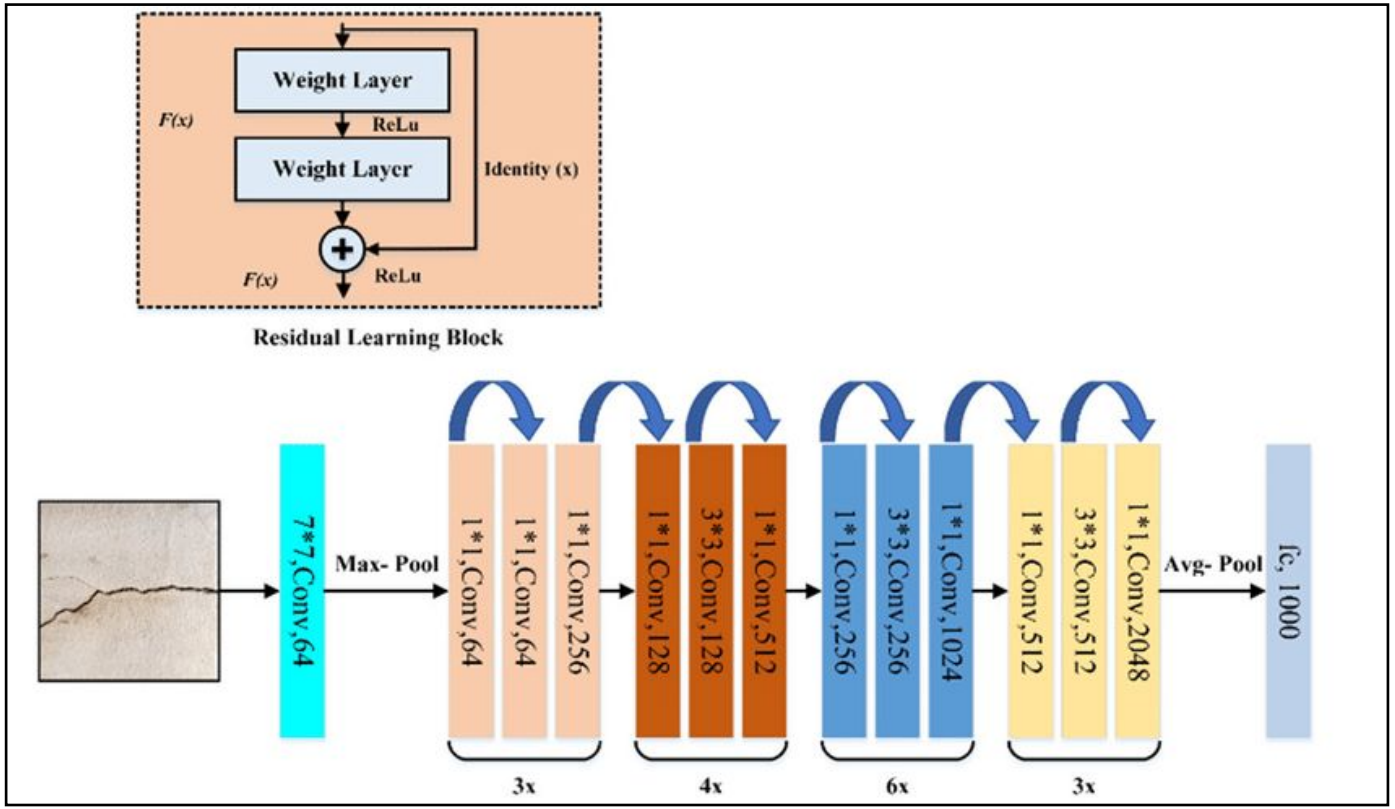


Conv Model	Recall	Precision	F1 score	Balanced Accuracy
VGG16	0.5527	0.5567	0.5433	0.5247

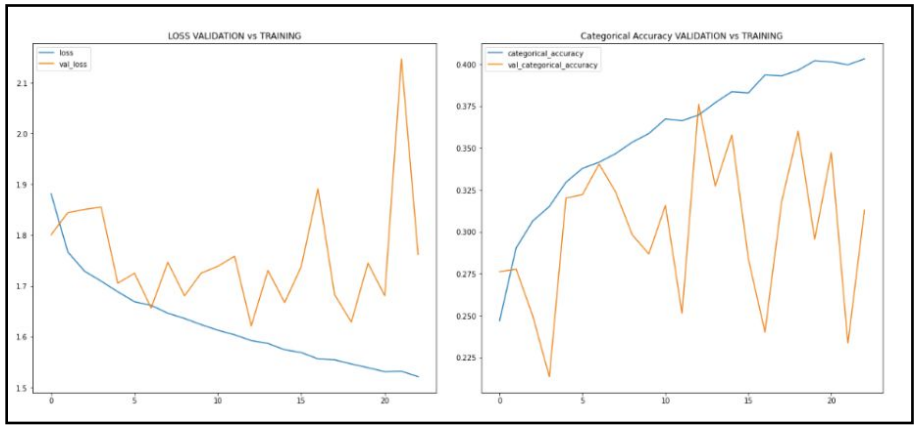
ResNet-50 with ImageNet weights

- A key innovation in the ResNet architecture was the residual module and its skip connection.
- The residual module is a block of two convolutional layers with the same number of filters and a small filter size.
- The output of the second layer is added with the input to the first convolutional layer. The input to the module is added to the output of the module which is called a shortcut connection.
- Like the VGG-16 model, transfer learning was implemented using the ImageNet weights for the ResNet-50 model and was trained after freezing the penultimate layers.
- The ResNet50 architecture with ImageNet weights achieved a training categorical accuracy of 36.96 % and a validation categorical accuracy of 37.61 % before plateauing.
- When evaluated using the test dataset, the model produced a weighted F1 score of 35.46 % and a balanced accuracy score of 30.24 %.

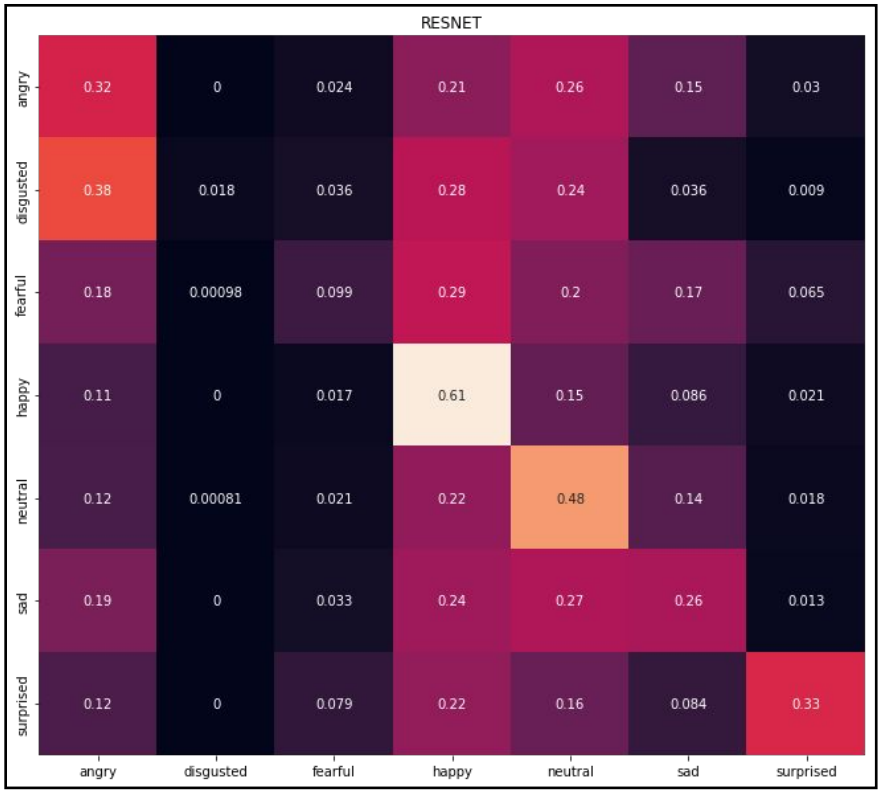
ResNet-50 block ConvNet Architecture



ResNet-50 ConvNet Architecture



Conv Model	Recall	Precision	F1 score	Balanced Accuracy
RESNET50	0.3735	0.3848	0.3546	0.3024



Evaluation Report

Conv Model	Recall	Precision	F1 score	Balanced Accuracy
Generic	0.6473	0.6493	0.6438	0.6118
VGG Block	0.6509	0.6527	0.6495	0.6226
INCEPTION Block	0.5139	0.5004	0.4994	0.4662
RESNET50	0.3735	0.3848	0.3546	0.3024
VGG16	0.5527	0.5567	0.5433	0.5247

Model Deployment

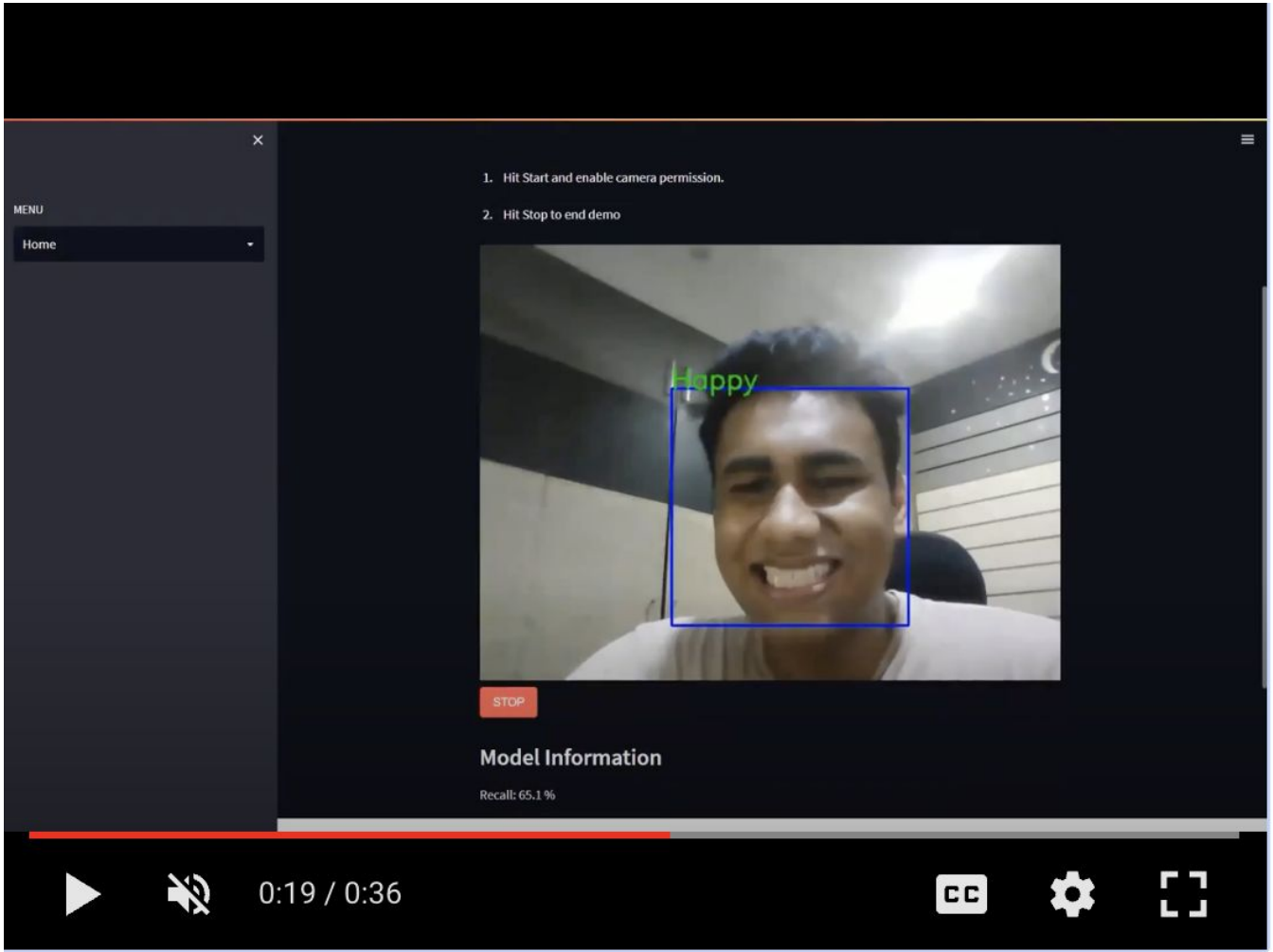
- For model deployment, OpenCV's Haarscascade Face Frontal file was used to capture the real-time video feed to detect and extract faces from video for emotion classification.

- Once the model was successfully executable, it was deployed into a web application for a usable interface. Streamlit library was used to build the front-end for the application.

- After running saliently on the local host, the model was deployed in the cloud using Heroku's platform and Streamlit Cloud for remote access.

- Remote access was observed to have a considerably long loading time as the compressed application size crossed the slug size soft limit of 300 megabytes.

Demo Snippet



Conclusions

- Thus during the course of the project, five models were built and compared to effectively identify student emotions using the FER-2013 dataset. The best classifying model, ConvNet with VGG blocks, had an F1 score of 65 % and a balanced accuracy score of 62.2 %.

- It was observed from the confusion matrix that the model could effectively classify emotions like Happy, Angry, Surprised and Neutral while emotions like Fear and Disgusted were classified with some difficulty. However, this issue only posed mild concern as those emotions were considered irrelevant in the context of digital learning.

- The model was used to create a web application to access the model and was deployed as a model on the cloud as an end-to-end solution.

- Further improvements that can be sought to be made are the inclusion of images with different levels of illumination and using sophisticated tools to make the web application lighter to suit the parallel processing of multiple video snippets for a real-life classroom environment.

References

1. How to Develop VGG, Inception and ResNet Modules from Scratch in Keras by Jason Brownlee
2. Jason Brownlee, “Imbalanced Classification with Python: Better Metrics, Balance Skewed Classes, Cost-Sensitive Learning”
3. Emotion Detection using Convolutional Neural Networks and OpenCV | Keras | Realtime

Thank You