Facial Expression Recognition with CNN

DONE BY:

1. MANOHAR VIRATI (17311A1210)
2. NIKITHA KATAKAM (17311A1215)
3. DHEERAJ PERUMANDLA (17311A1232)
4. SOWMYA RACHA (17311A1237)

Convolutional Neural Network (CNN)

- CNNs are neural networks which are sensitive to spatial information. They are capable of recognizing complex shapes and patterns in a given image and are used for a wide variety of image classification algorithms.

About the Project

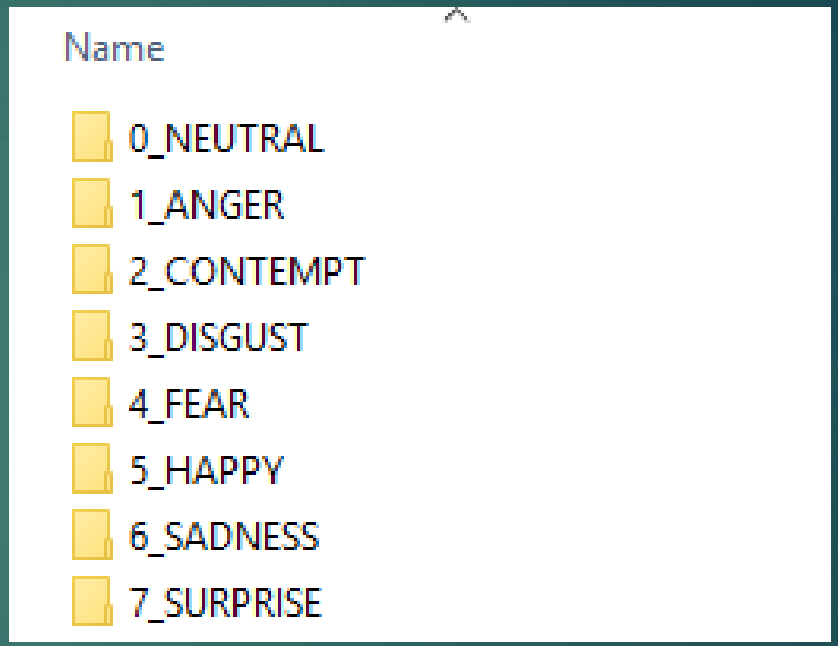
- In this project we have built a CNN which is capable of classifying eight facial emotions (Neutral, Anger, Contempt, Disgust, Fear, Happiness, Sadness and Surprise) with high degree of accuracy.
- The programs are written to run on Python 3.7
- A virtual environment(opencv-env)was created and required packages were downloaded with the help of ANACONDA NAVIGATOR AND PROMPT.
- PYCHARM was used in this project.
- The LBP Cascade Classifier for frontal face detection available in OpenCV has been used to detect faces in images.
- The CNN for detecting facial emotions was built using Keras.
- The images used in this project are from Cohn-Kanade (CK and CK+) database

Program Modules

faceDetectoLayer.py	Contains the code which leverages OpenCV's LBP Cascade Classifier for frontal facial detection in order to detect the faces in a given image
ckPlusLayer.py	This module is used to navigate the CK+ dataset, detect the faces in the images using the faceDetectorInterface.py module and store them in separate folders categorized by their emotions.
dataLayer.py	Controls the work done by the ckPlusInterface.py and consolidates the labeled faces and stores them into data.npy and labels.npy files.
emotionRecognitionNetwork.py	Designs and trains the CNN according to specifications, stores the network into a file and provides functionality to use the trained network to make predictions.
cnnLayer.py	Used to specify the CNN network structure, data pre-processing and exposes interfaces to enable other applications to use the network for predictions.
applicationInterface.py	Contains application which detects faces from the video input through the webcam and classifies its expressions.
masterControl.py	User interface which runs the other modules.

Data Pre-processing

- The images in the CK+ database were scanned to extract the portion of the image which contains the face.
- The detected faces were resized into 100X100 images and stored into directories named after different emotions



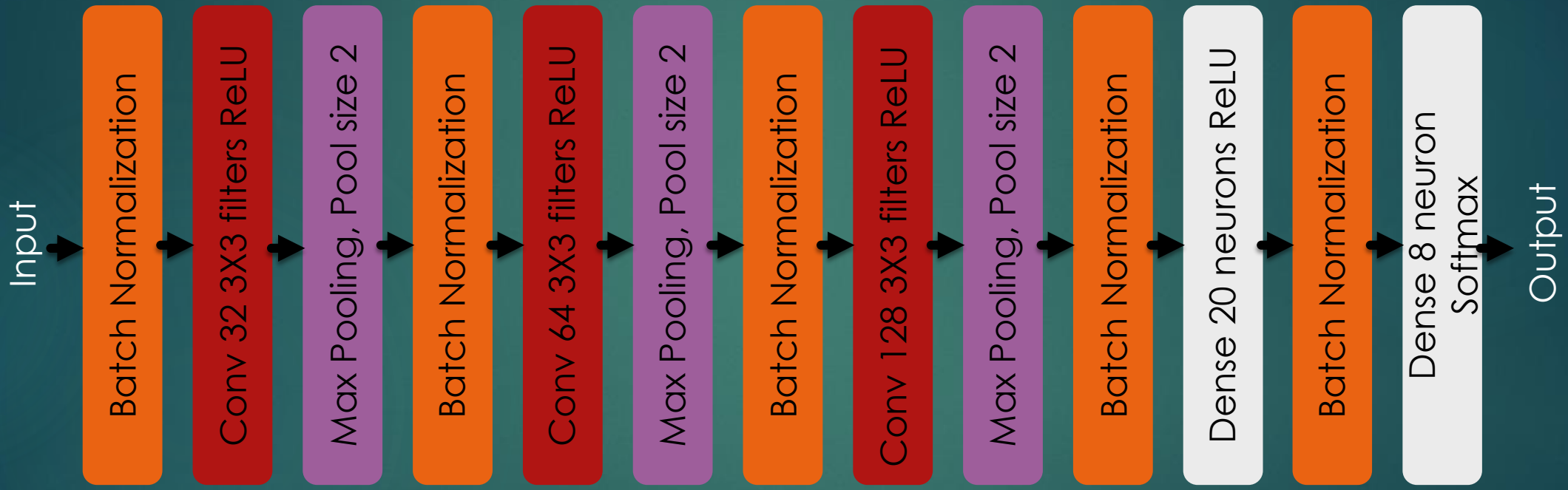
Data Pre-processing (continued)

- The images stored in the emotion was manually inspected to remove any flawed images.
- The images were then consolidated into the data.npy file and their corresponding labels were stored in the labels.npy file. During this process, emotions with poor representation were identified and additional images with minor amount of noise were added into the nympy files to ensure that every emotion is equally represented for training.

Emotion Recognition Network

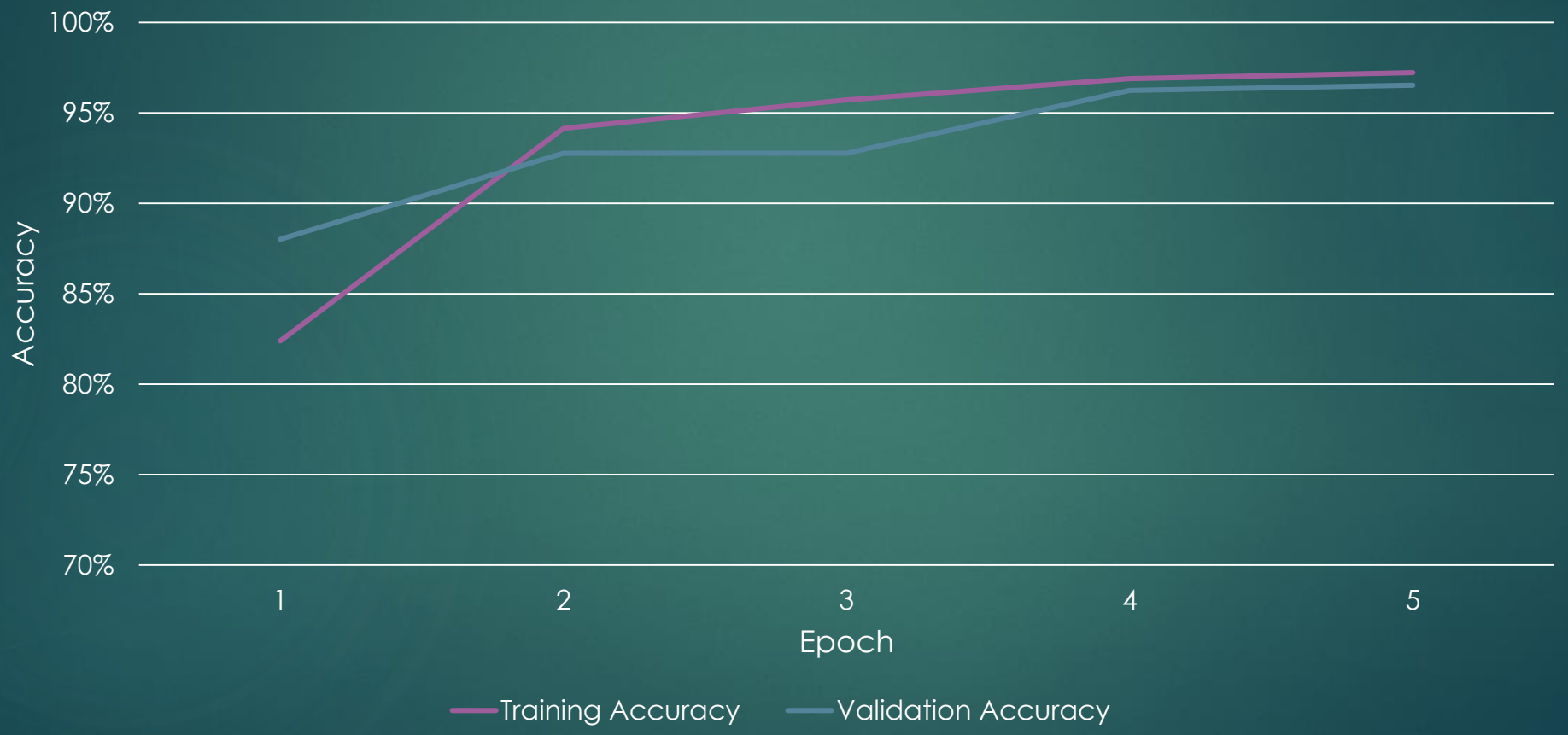
- The data was imported from data.npy and labels.npy files, randomly shuffled and split into a training dataset containing 80% of the images, a testing dataset containing 10% of the images and the validation dataset containing 10% of the images.
- The CNN contains 32 filters in the first layer with ReLU, 64 filters on the second layer and 128 filters on the third layer. All layers use ReLU activation. Max Pooling was done after every CNN layer to improve speed of the network.
- A Dense network was built over the CNN. It contains 20 neurons on the first layer with ReLU activation and 8 neurons on the second layer with Softmax activation.
- Batch normalization was done before every layer in the network.

Emotion Recognition Network (continued)



Network Structure

Emotion Recognition Network (continued)



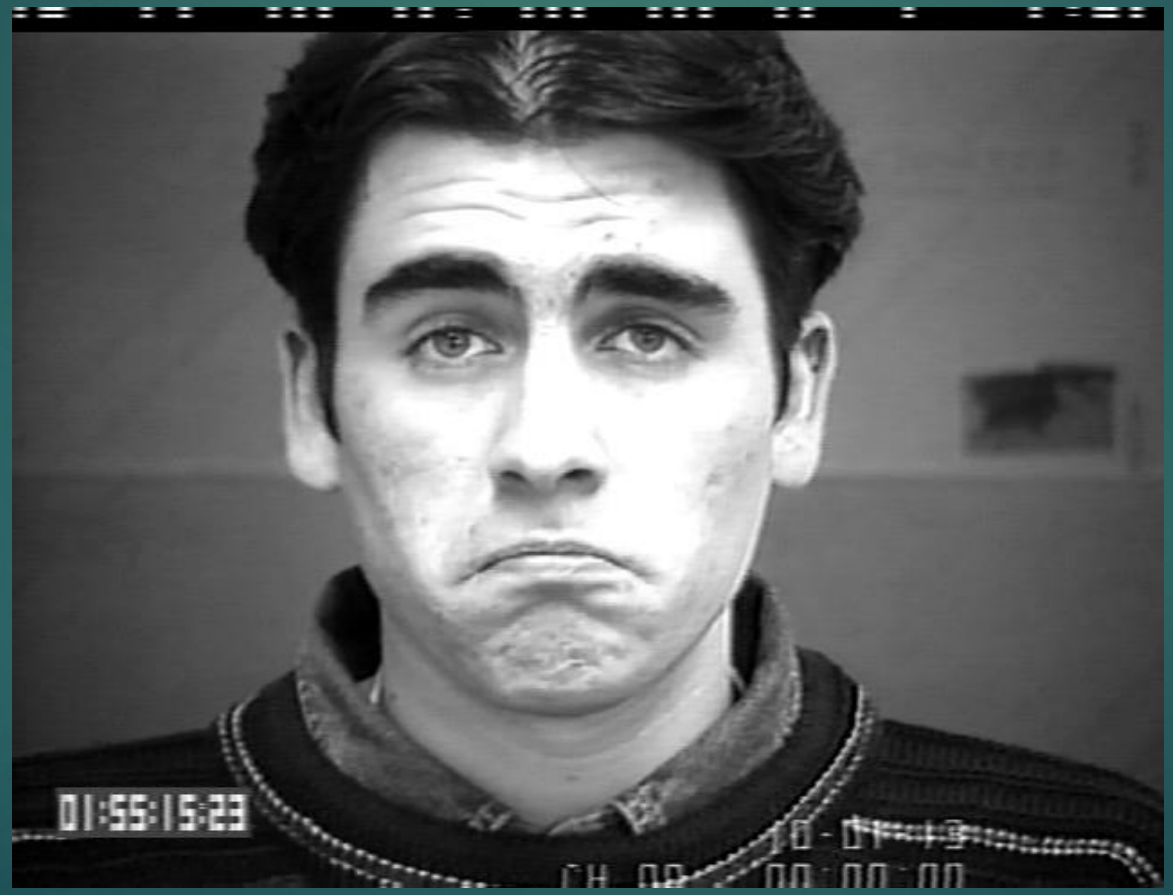
Emotion Recognition Network (continued)

Actual \ Pred	0_NEUTRAL	1_ANGER	2_CONTEMPT	3_DISGUST	4_FEAR	5_HAPPY	6_SADNESS	7_SURPRISE
0_NEUTRAL	410	7	0	14	2	7	16	15
1_ANGER	0	483	0	1	0	0	1	1
2_CONTEMPT	0	0	442	0	0	0	0	0
3_DISGUST	0	0	0	436	0	0	0	1
4_FEAR	0	0	0	0	437	0	0	0
5_HAPPY	0	1	0	0	0	441	2	5
6_SADNESS	0	0	0	0	0	0	494	0
7_SURPRISE	1	0	0	2	0	0	2	468

Confusion Matrix with 95.96% accuracy

Demo Anger ©Jeffrey Cohn

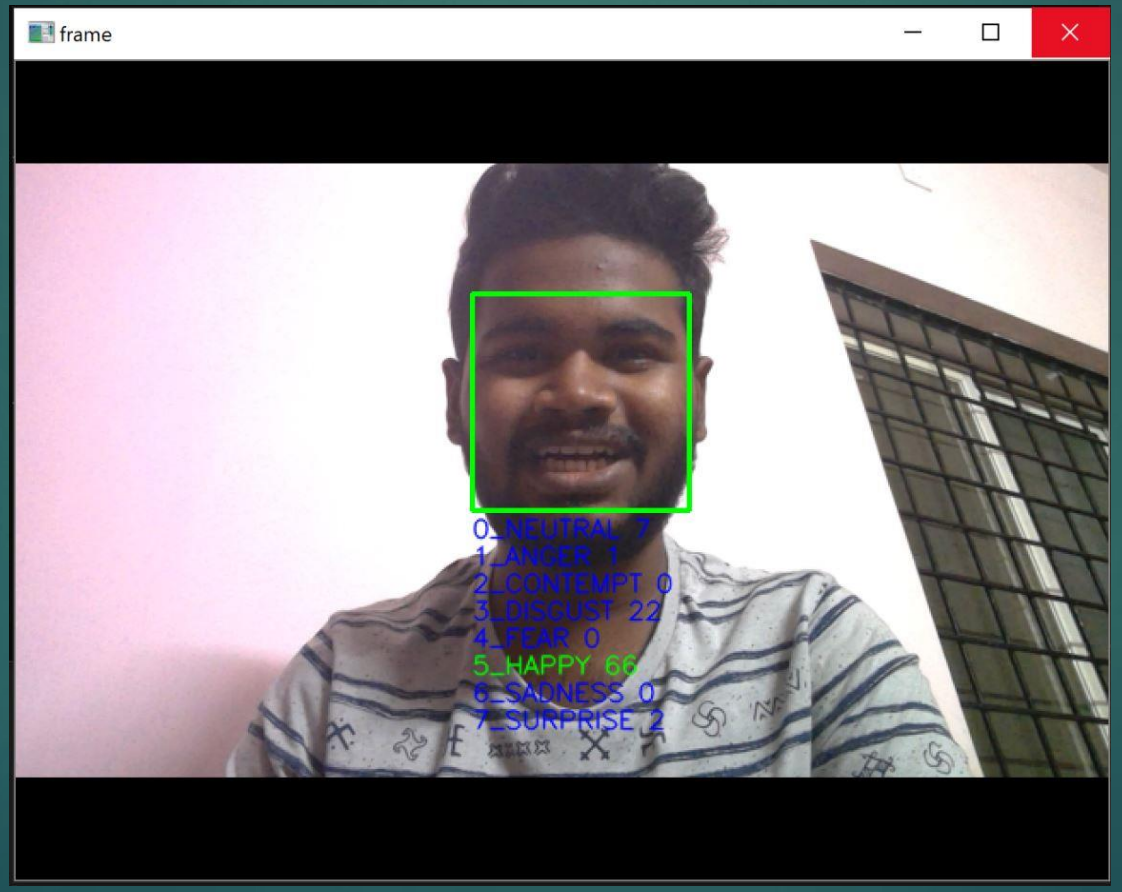
Demo Sadness ©Jeffrey Cohn



Demo Fear ©Jeffrey Cohn

Demo Happiness ©Jeffrey Cohn

OUTPUT:


frame
0_NEUTRAL 7
1_ANGER 1
2_CONTEMPT 0
3_DISGUST 22
4_FEAR 0
5_HAPPY 66
6_SADNESS 0
7_SURPRISE 2

Thank You