Logo

Description automatically generated

**Program: Computer Engineering and Software Systems**

***Course Code: CSE485***

***Course Name: Deep Learning***

***Name: Moataz Alaa Eldeeb***

***ID: 19P6238***

***Under Supervision on***

**Dr. Alaa Hamdy**

***Ain Shams University***

***Faculty of Engineering***

Deep Learning Project

# Project Description

Face Emotion Detection

The data consists of 48x48 pixel grayscale images of faces. The faces have been automatically registered so that the face is more or less centered and occupies about the same amount of space in each image.

The task is to categorize each face based on the emotion shown in the facial expression into one of seven categories (0=Angry, 1=Disgust, 2=Fear, 3=Happy, 4=Sad, 5=Surprise, 6=Neutral). The training set consists of 28,709 examples and the public test set consists of 3,589 examples.

It is one of the more challenging datasets with human-level accuracy only at 65±5% and the highest performing published works achieving single model 73.2% test accuracy.



# Data Splitting

Data is split into training (28,709 images), validation (3,589), testing (3,589). Cross validation is used to ensure that testing set is not seen by the model during training. It is test after finishing training.

# Data Augmentation

Data augmentation is a technique of artificially increasing the training set by creating modified copies of a dataset using existing data. It includes making minor changes to the dataset or using deep learning to generate new data points. Resizing data was done to make sure it fit the transfer learning algorithms.

A picture containing text, screenshot, software, multimedia

Description automatically generated

I used width/ height shift range to be 10%, zoom range to be 10%, radiation range to be 10 degrees, and with horizontal flip. After all this, divide each pixel by 255 to normalize data before training.

# Models Used

I used 2 transfer learning architectures which are resnet50 and vgg16. But with trained data on VGGFace data all with batch size 128, Stochastic gradient descent and cross entropy loss. Using VGGFace instead of imagenet trained weights as imagenet weights was tried but the accuracy saturated near 40% after 30 epochs.

After this I used my own model inspired by vgg-architecture

VGGFace dataset:



## Resnet50

### Model Description

Two models were trained using resnet50, one with class weights because fer2013 data is unbalanced, one without. Freezing the first 170 layer with the same weights as vggface data. And using dropout before each fully connected layer for regularization.

A screen shot of a computer program

Description automatically generated with low confidence

### Training curves

With Class weights:

Final test accuracy = 68.1

Final validation accuracy = 69.2

A picture containing text, screenshot, plot, line

Description automatically generated

A picture containing text, line, diagram, screenshot

Description automatically generated

Without Class weights:

Final test accuracy = 70.65

Final validation accuracy = 73.43

A picture containing text, line, plot, screenshot

Description automatically generated

A picture containing text, line, screenshot, plot

Description automatically generated

## VGG16 Model

### Model Description

VGG-16 architecture was used with pretrained VGGFace weights. Here I only trained one model with weighted classes to balance data.

A picture containing text, screenshot, software, operating system

Description automatically generated

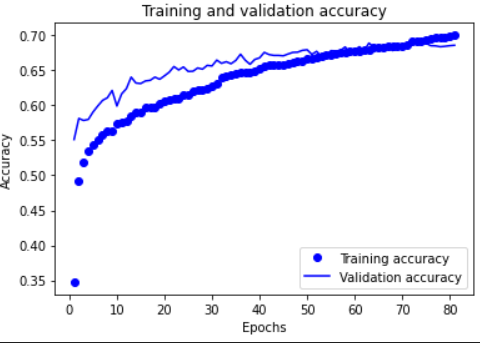
### Training Curves

Final test accuracy = 0.69

Final validation accuracy = 0.7001

A picture containing text, screenshot, line, plot

Description automatically generated



# Live Testing App

|  |  |
| --- | --- |
| A screenshot of a computer  Description automatically generated | A screenshot of a computer  Description automatically generated |
|  |  |