Emotion Detection using Deep Convolutional Neural Network

Presented By:
Muhammad Hammad & Elliott Barnes

Overview:

- Problem Statement
- Proposed Solution
- Data and Technology
- Results
- Demo
- References

1.

Problem Statement

How do we classify emotions?

Problem Statement:

- Classify between two unique emotions:
  - Happy 
  - Neutral 

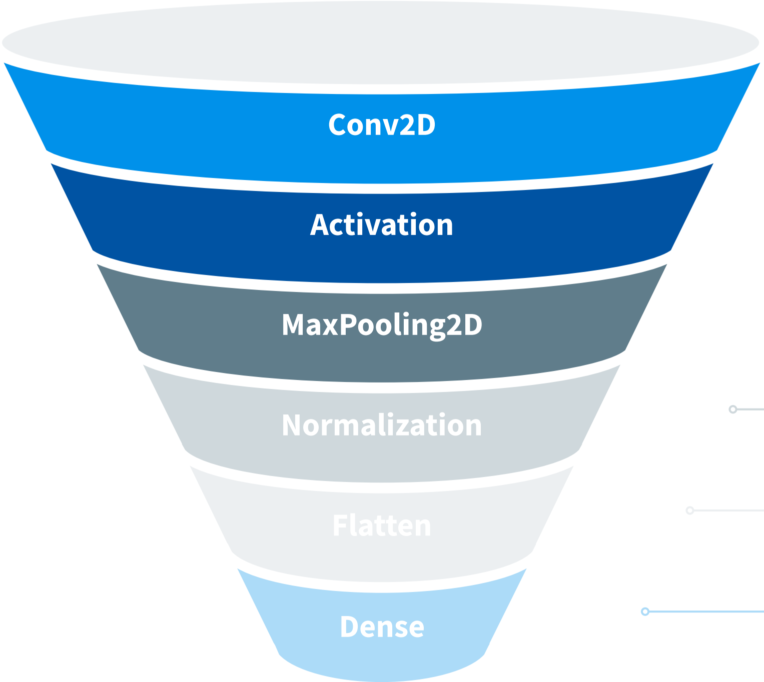
- Applications:
  - Apps that take pictures on smiles
  - Psych research

2.

Proposed Solution

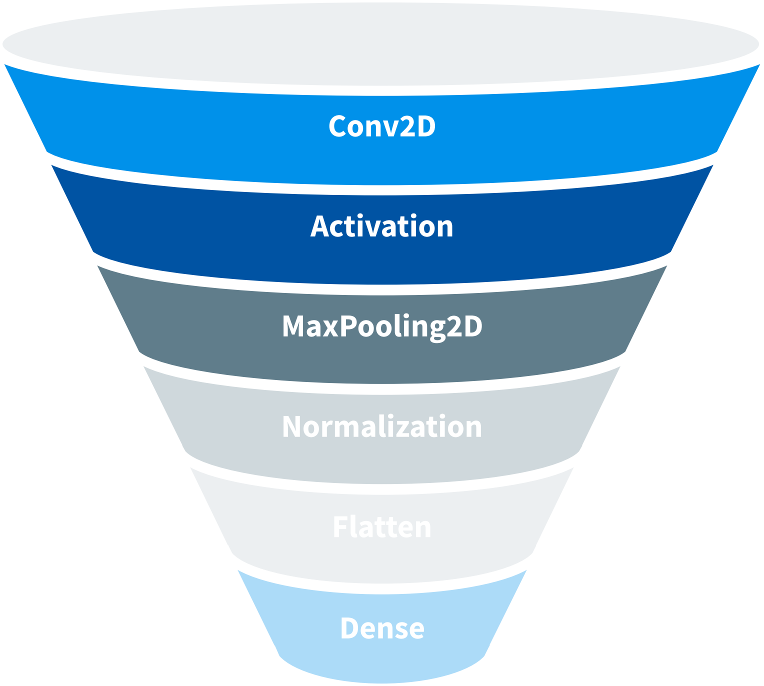
Deep Convolutional Neural Network

General Model



- Convolution Layer (filters = 32/64/128, kernel = 3x3)
- Activation Layer (function = ReLU/softmax)
- Max Pooling Layer (size = 2x2)
- Batch Normalization Layer [0,1]
- Flatten Layer
- Dense Layer (units = 64/32/2)

General Model

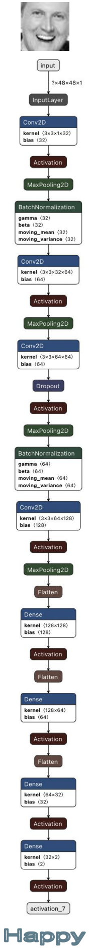


Proposed Solution:

Deep Convolutional Neural Network Model

- **Model**
  - Sequential Model
  - 4 Encoding Blocks
  - 4 Decoding Blocks

- **Training Details**
  - 50 Epochs
  - 32 Batch Size
  - 48x48 Pixels

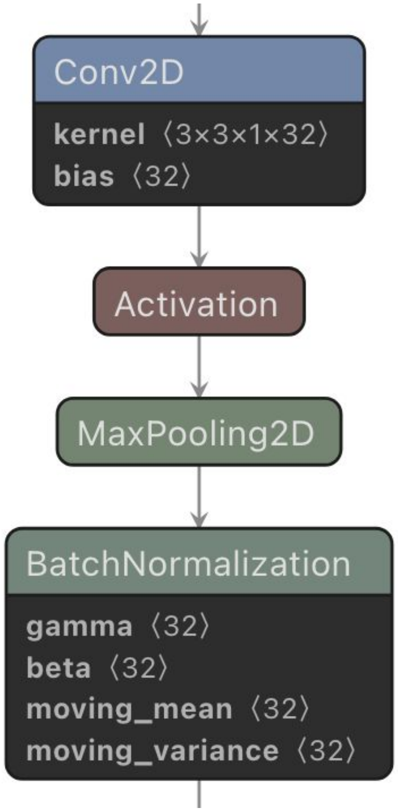



```python
# Create the sequential model for the CNN
model_int = models.Sequential()

model_int.add(Conv2D(32, (3, 3), input_shape=(48, 48, 1)))
model_int.add(Activation("relu"))
model_int.add(MaxPool2D((2, 2)))
model_int.add(BatchNormalization())

model_int.add(Conv2D(64, (3, 3)))
model_int.add(Activation("relu"))
model_int.add(MaxPool2D((2, 2)))

model_int.add(Conv2D(64, (3, 3)))
model_int.add(Dropout(0.1))
model_int.add(Activation("relu"))
model_int.add(MaxPool2D((2, 2)))
model_int.add(BatchNormalization())

model_int.add(Conv2D(128, (3, 3)))
model_int.add(Activation("relu"))
model_int.add(MaxPool2D((2, 2)))

model_int.add(Flatten())
model_int.add(Dense(128))
model_int.add(Activation("relu"))

model_int.add(Flatten())
model_int.add(Dense(64))
model_int.add(Activation("relu"))

model_int.add(Flatten())
model_int.add(Dense(32))
model_int.add(Activation("relu"))

model_int.add(Dense(NUM_OF_EMOTIONS))
model_int.add(Activation('softmax'))
```

Proposed Solution:

Encoding Block 1

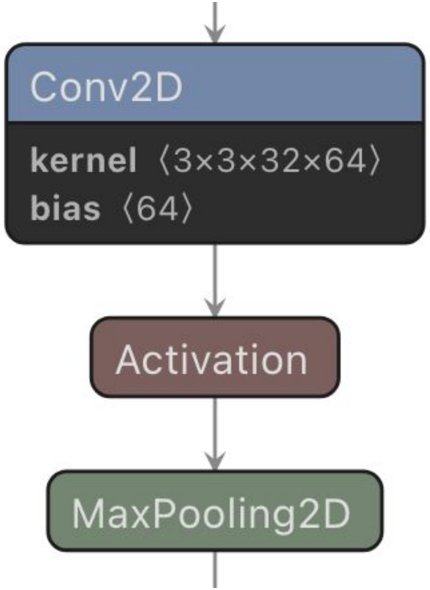
- The first convolutional layer learns 32 filters, and uses a 3x3 mask (no need for bigger, our input images are only 48x48)
- Then, we use the Rectified Linear Unit (ReLU) activation function
- Reduce the spatial dimensions of the output with max pooling, using a 2x2 mask
- And normalize the output with Batch Normalization



Proposed Solution:

Encoding Block 2

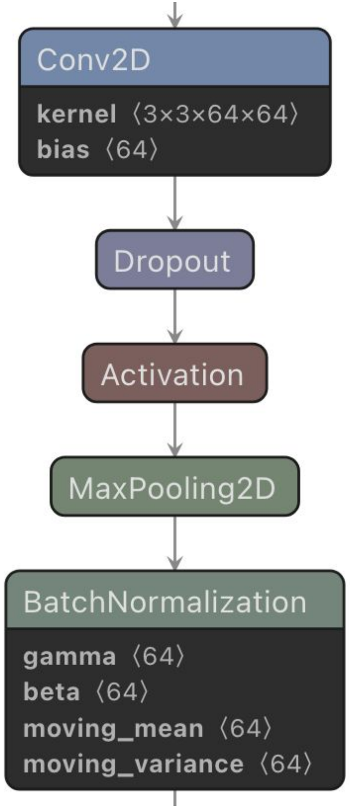
- The second convolutional layer learns 64 filters, and uses a 3x3 mask
- Then, we use the Rectified Linear Unit (ReLU) activation function
- Reduce the spatial dimensions of the output with max pooling, using a 2x2 mask



Proposed Solution:

Encoding Block 3

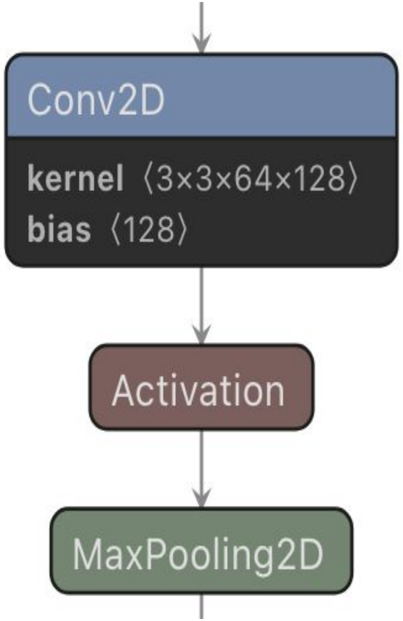
- ➢ The third convolutional layer learns 64 filters, and uses a 3x3 mask
- ➢ We added a Dropout layer with a dropout rate of 0.1 to reduce overfitting
- ➢ Then, we use the Rectified Linear Unit (ReLU) activation function
- ➢ Reduce the spatial dimensions of the output with max pooling, using a 2x2 mask
- ➢ And normalize the output with Batch Normalization



Proposed Solution:

Encoding Block 4

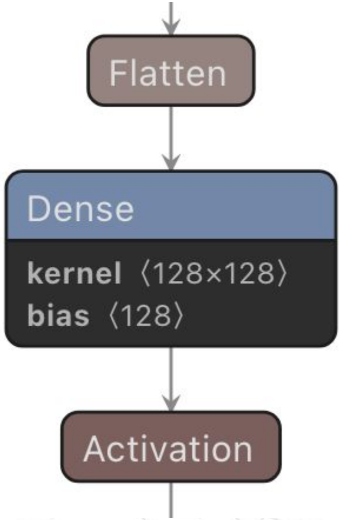
- The fourth convolutional layer learns 128 filters, and uses a 3x3 mask
- Then, we use the Rectified Linear Unit (ReLU) activation function
- Reduce the spatial dimensions of the output with max pooling, using a 2x2 mask



Proposed Solution:

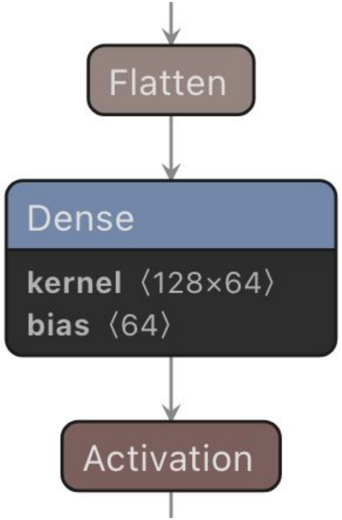
Decoding Block 1

- ➢ Flatten layer
- ➢ Dense 128 units
- ➢ Activation (ReLU)



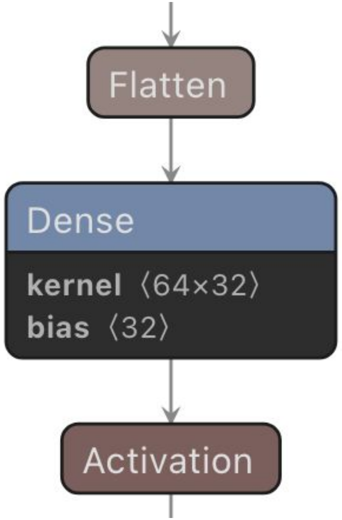
Decoding Block 2

- ➢ Flatten layer
- ➢ Dense 64 units
- ➢ Activation (ReLU)



Decoding Block 3

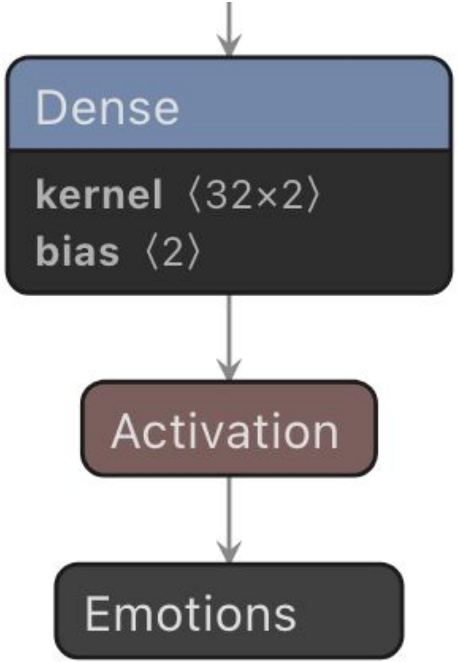
- ➢ Flatten layer
- ➢ Dense 32 units
- ➢ Activation (ReLU)



Proposed Solution:

Decoding Block 4

- Dense 2 units
- Activation (ReLU)
- Output one of two emotions



Proposed Solution:

- **Optimizer**
  - Adam
    - Works well on large datasets
    - Can handle sparse gradients on noisy datasets
    - Computationally efficient

- **Loss Function**
  - Binary Cross-Entropy
    - Works well for binary classification

3.

Data & Technology

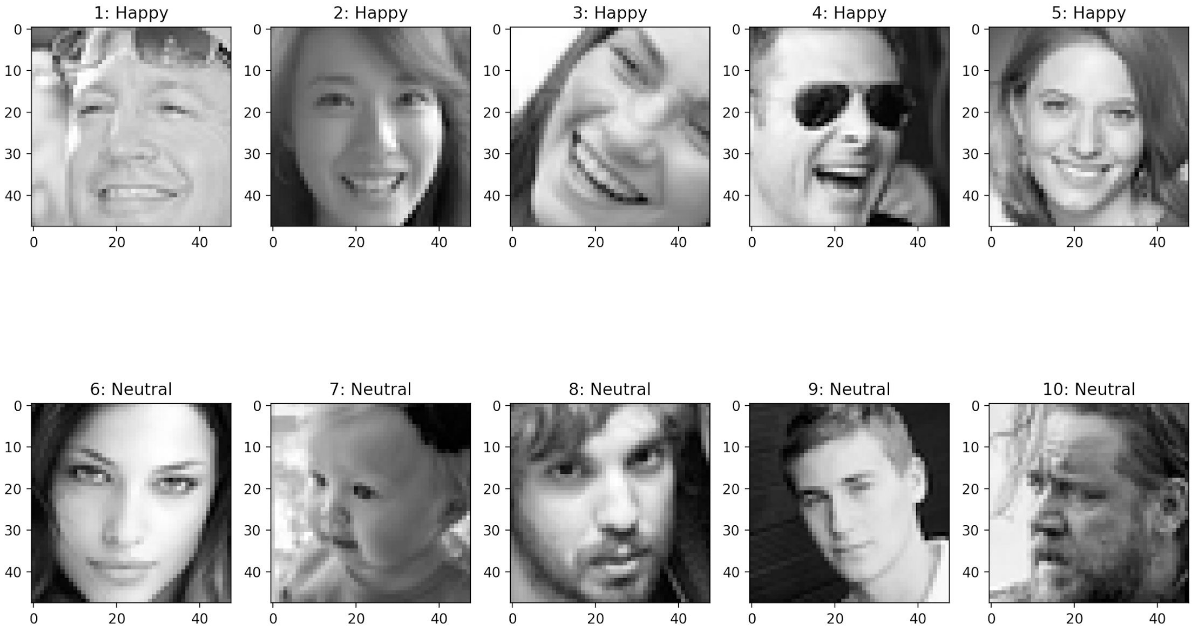
Data and Technology:

- **Technology**
  - Python
    - Tensorflow/Keras, Matplotlib, OpenCV, Pandas, and Numpy

- **Dataset**
  - Facial Expression Recognition Challenge Dataset - 2013
    - Downloaded from Kaggle
    - FER2013 Link

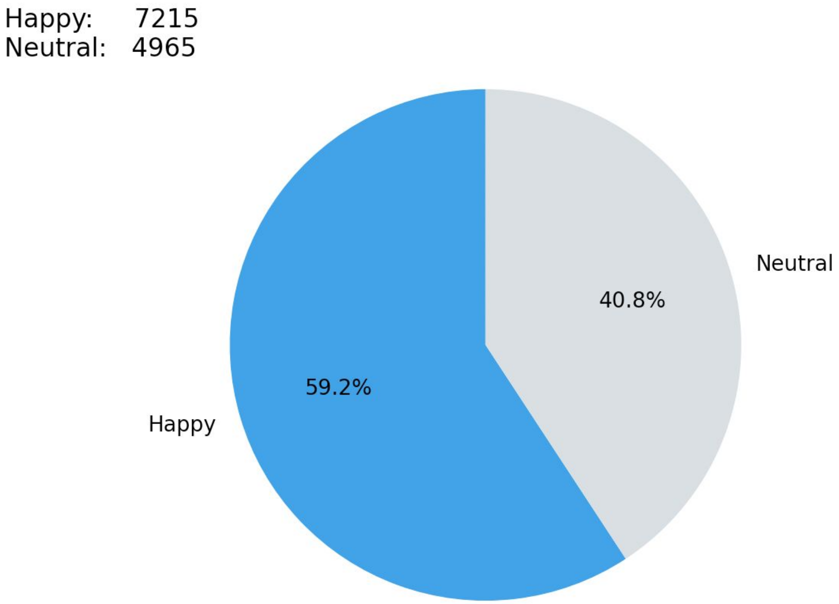
Data and Technology:

Random Images from Training Set



Data and Technology:

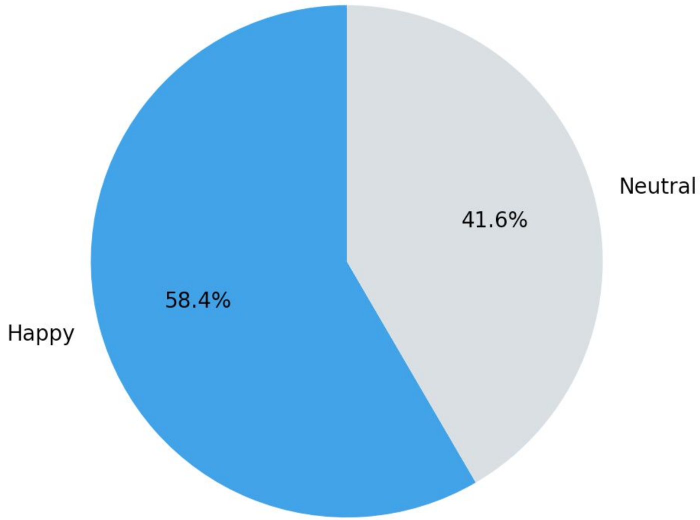
Training Set



Data and Technology:

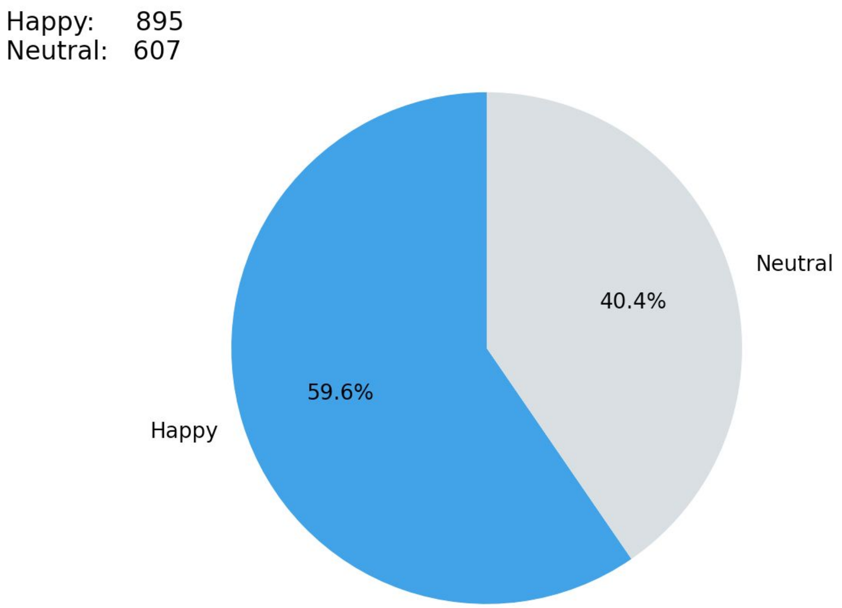
Test Set

Happy: 879
Neutral: 626



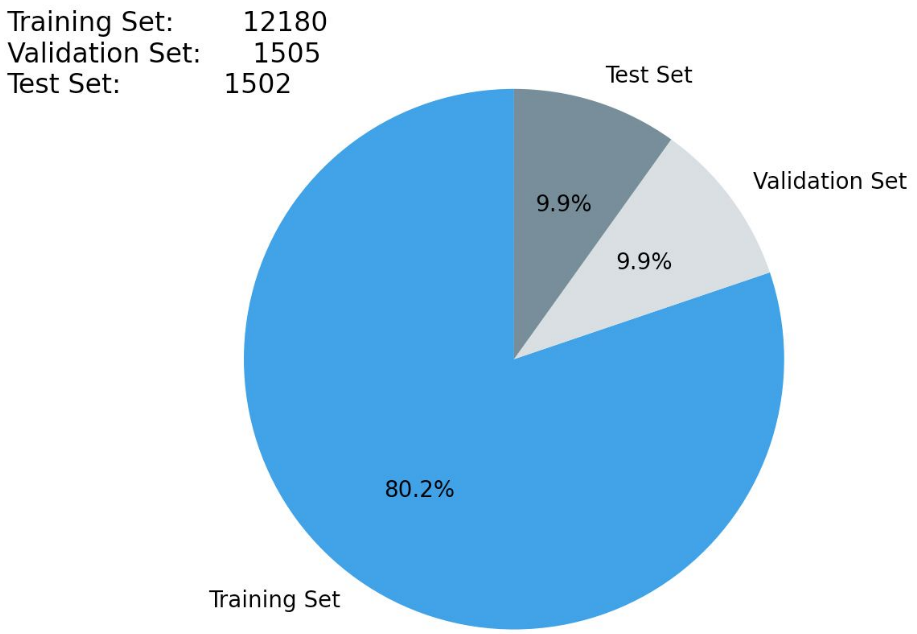
Data and Technology:

Validation Set



Data and Technology:

Distribution of all sets



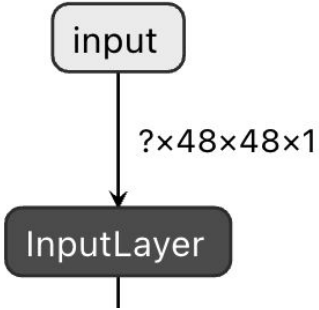
Data and Technology:

Data Augmentation

Expanded Dataset

By introducing transformations such as:

- ➢ Rotation
- ➢ Shear
- ➢ Zoom
- ➢ Horizontal Flip / Mirroring
- ➢ Width and Height Shift

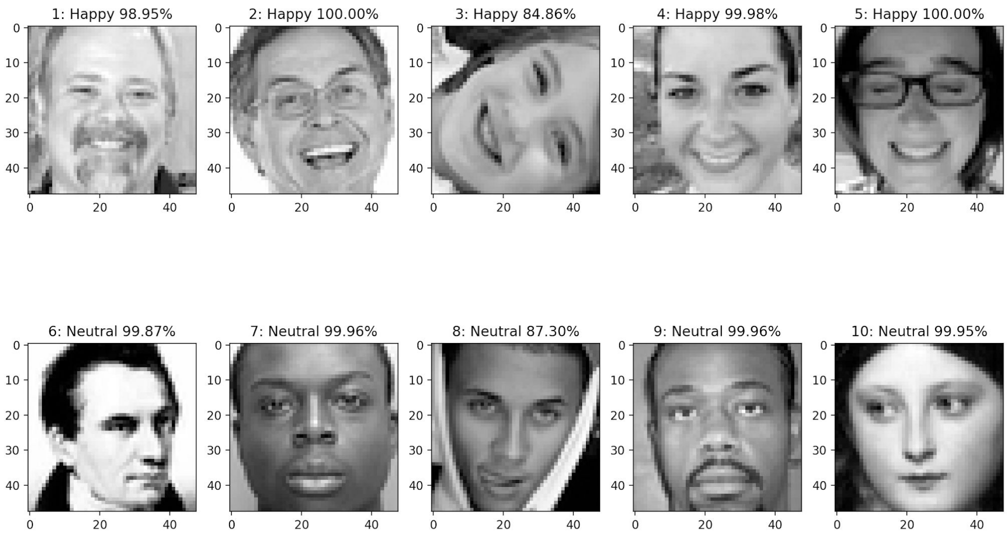


4.

Results

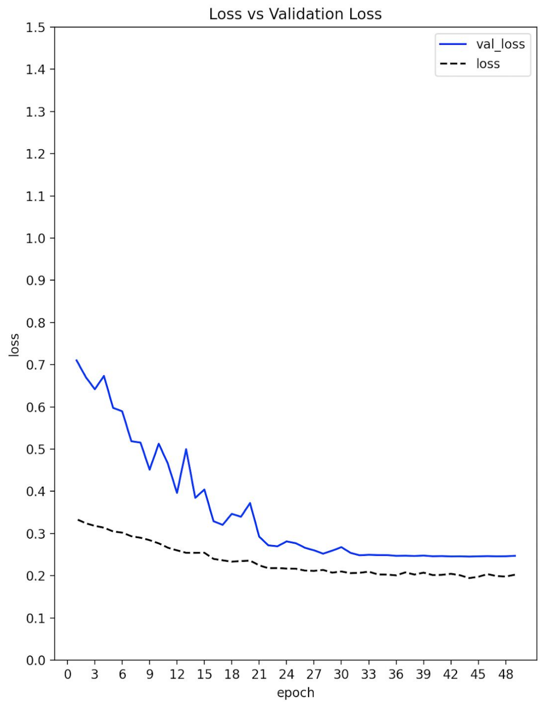
Results:

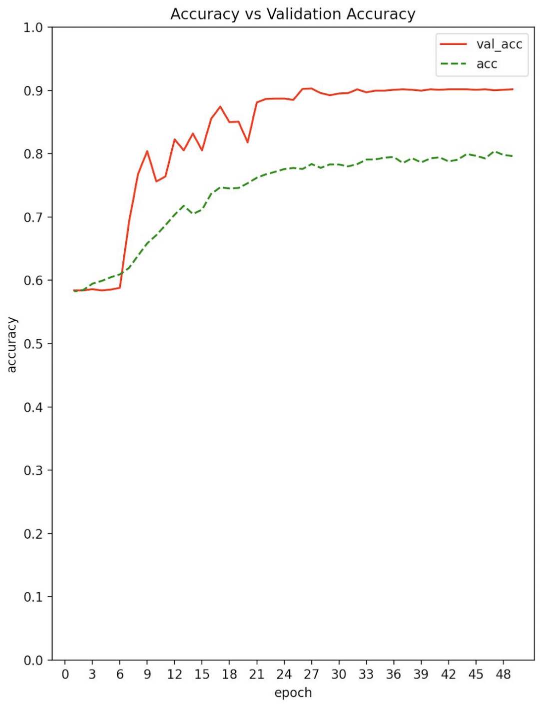
Figure of Predictions



Results:

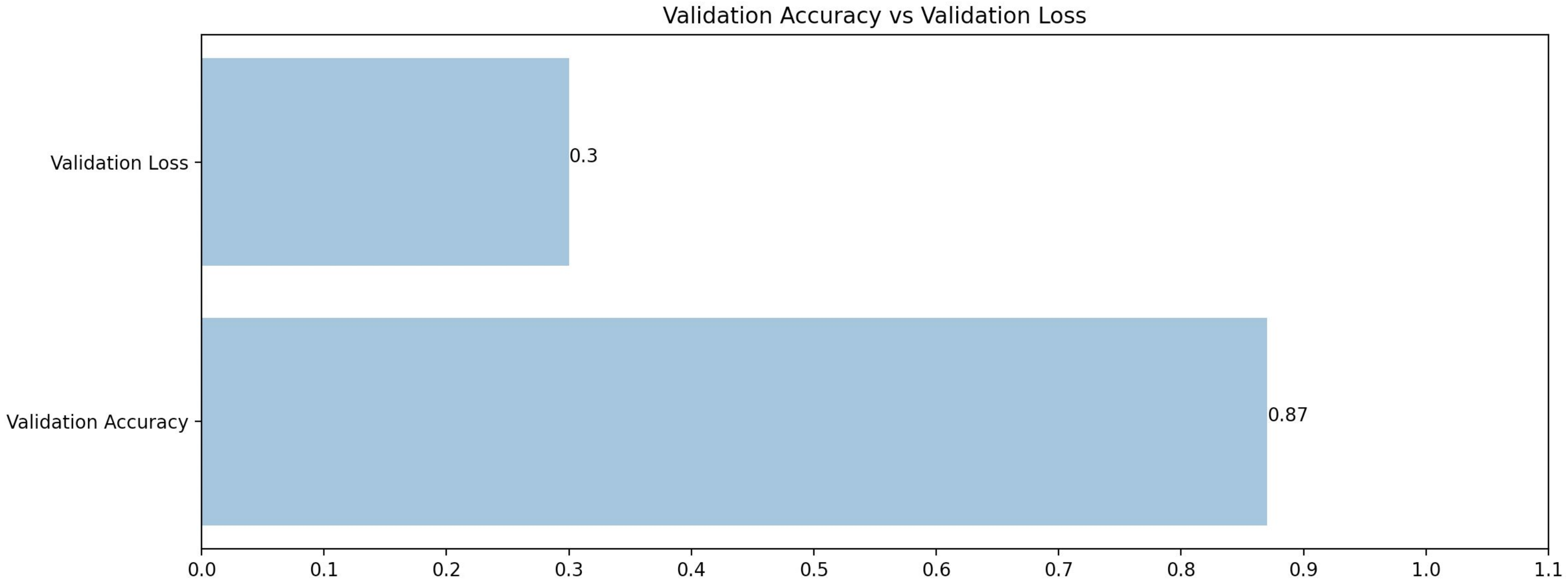
Loss and Accuracy





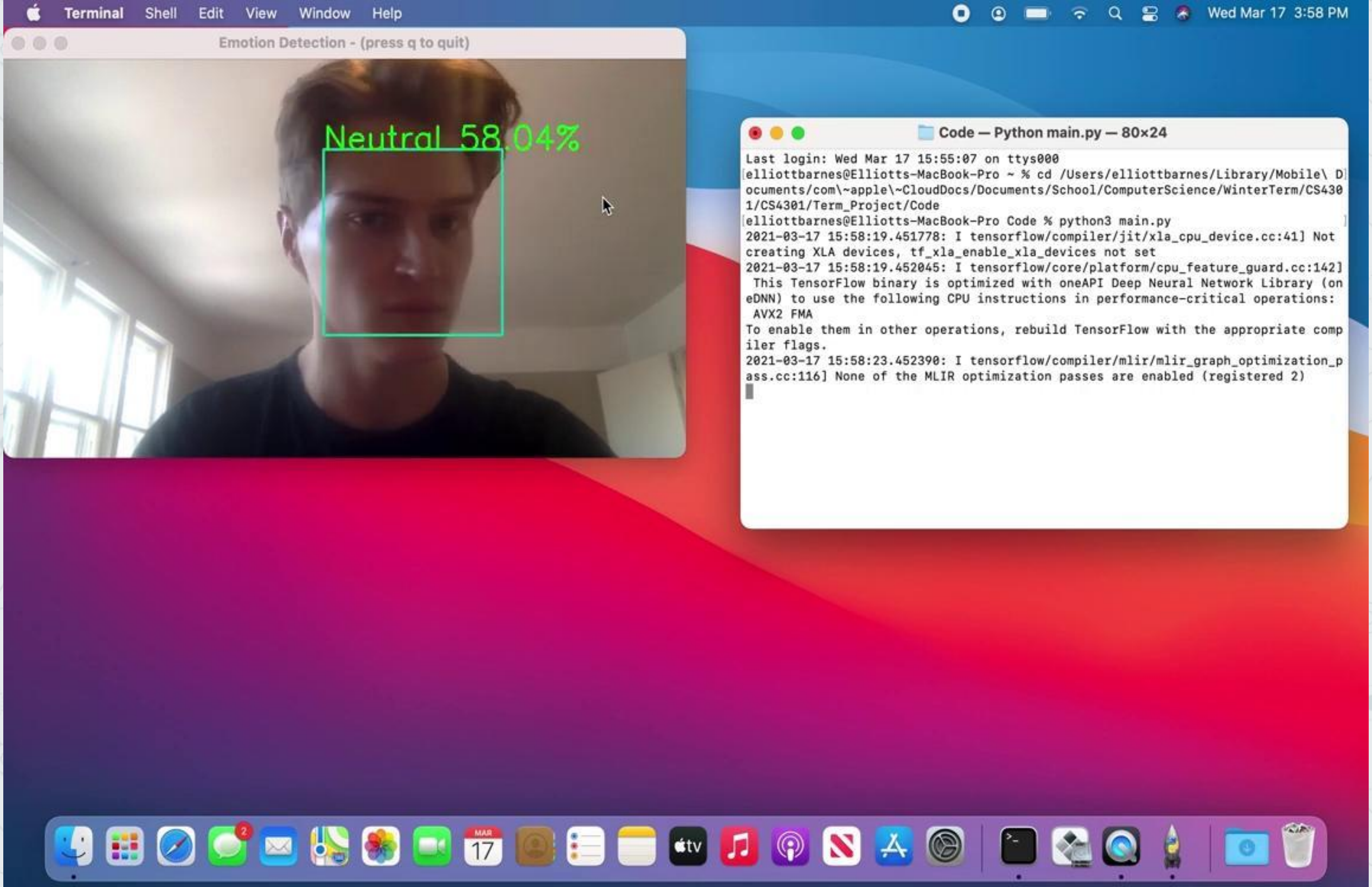
Results:

Our Test Loss vs Accuracy



5.

Demo


Terminal Shell Edit View Window Help
Wed Mar 17 3:58 PM
Emotion Detection - (press q to quit)
Neutral 58.04%
Code — Python main.py — 80×24
Last login: Wed Mar 17 15:55:07 on ttys000
elliottbarnes@Elliotts-MacBook-Pro ~ % cd /Users/elliottbarnes/Library/Mobile\ Documents/com\~apple\~CloudDocs/Documents/School/ComputerScience/WinterTerm/CS4301/CS4301/Term_Project/Code
elliottbarnes@Elliotts-MacBook-Pro Code % python3 main.py
2021-03-17 15:58:19.451778: I tensorflow/compiler/jit/xla_cpu_device.cc:41] Not creating XLA devices, tf_xla_enable_xla_devices not set
2021-03-17 15:58:19.452045: I tensorflow/core/platform/cpu_feature_guard.cc:142] This TensorFlow binary is optimized with oneAPI Deep Neural Network Library (oneDNN) to use the following CPU instructions in performance-critical operations: AVX2 FMA
To enable them in other operations, rebuild TensorFlow with the appropriate compiler flags.
2021-03-17 15:58:23.452390: I tensorflow/compiler/mlir/mlir_graph_optimization_pass.cc:116] None of the MLIR optimization passes are enabled (registered 2)

6.

References

References:

1. *Li, S., Xing, J., Niu, Z., Shan, S., & Yan, S. (2015). Shape driven kernel adaptation in Convolutional Neural Network for robust facial trait recognition. 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). doi:10.1109/cvpr.2015.7298618*

2. *Xiaoguang Chen, Xuan Yang, Maosen Wang, & Jiancheng Zou. (2017). Convolution neural network for automatic facial expression recognition. 2017 International Conference on Applied System Innovation (ICASI), 814-817.*

3. *Matsugu, M, Mori, K, Mitari, Y, & Keneda, Y. (2003). Facial expression recognition combined with robust face detection in a convolutional neural network. Proceedings of the International Joint Conference on Neural Networks, 2003, 3, 2243-2246 vol.3.*

4. *Dataset - Facial Expression Recognition Challenge Dataset - 2013*
    a. *https://www.kaggle.com/c/challenges-in-representation-learning-facial-expression-recognition-challenge/data*
5. *Keras*
    a. *https://keras.io/*
6. *Open CV*
    a. *https://opencv.org/*
7. *Slide Template*
    a. *https://www.slidescarnival.com/*
8. *Model Visualization*
    a. *https://netron.app/*

Questions & Answers