2021

Facial Emotion Recognition

PROJECT REPORT

NGOC PHAN

BIJESH PATEL VACHANNI

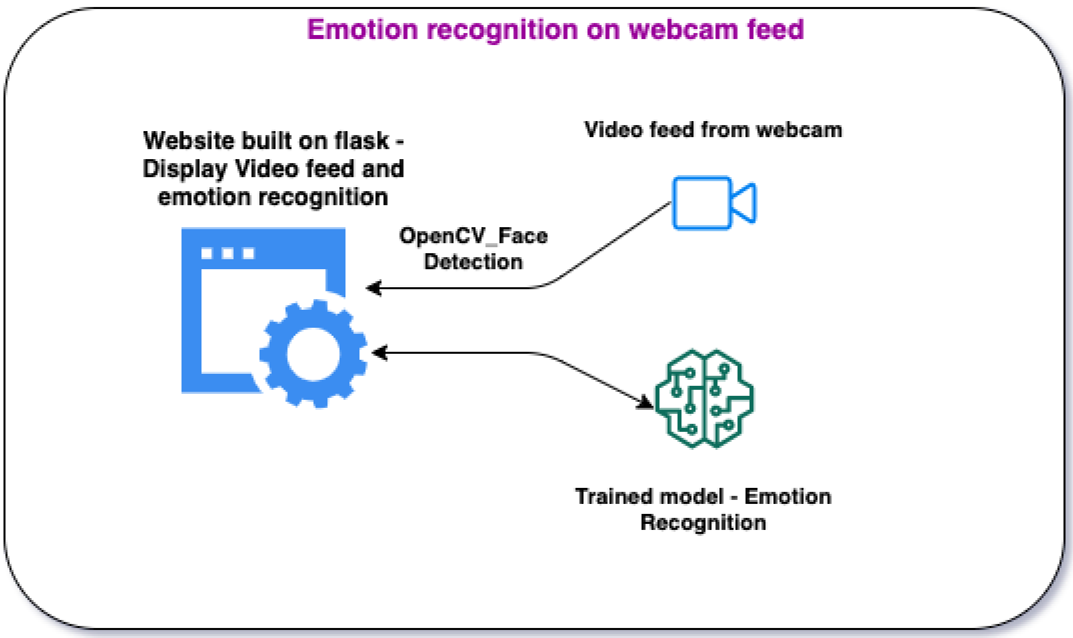
NAGA SUMANTH VANKADARI

April 23, 2021

On my honor as a UNT student, I have neither given nor received unauthorized assistance on this work.

Abstract

Emotion recognition of a human face from a photo/video frame is widely used in consumer industry and other industries to gauge human behavior, attentiveness and analyze the emotion. In our project, we worked on building a model that could predict emotion on the human face. Trained model is integrated into a website where the human face is captured from a webcam video feed and live emotion detection is captured.



Data Specification

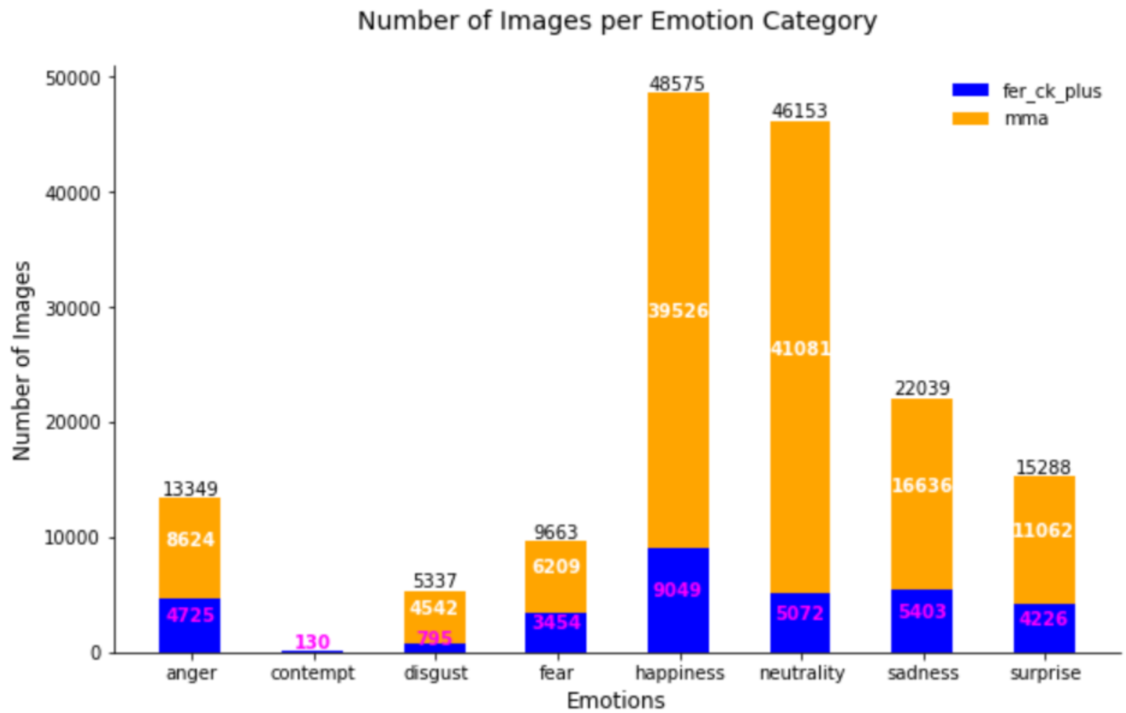
Two facial emotion expression datasets were collected on Kaggle:

1) ***fer_ckplus_kdef***
   - Url: https://www.kaggle.com/sudarshanvaidya/corrective-reannotation-of-fer-ck-kdef
   - Description: The dataset contains 32,854 images with 8 emotions categories – anger, contempt, disgust, fear, happiness, neutrality, sadness and surprise. All images contain grayscale human faces (or sketches). Each image is 224 x 224 pixel grayscale in PNG format.

2) ***MMA Facial Expression***
   - Url: https://www.kaggle.com/mahmoudima/mma-facial-expression
   - Description: The dataset has 127,680 images (both color and grayscale), and contains tree directories for training, validation, and test. Each directory contains seven sub-directories

corresponding to seven facial expression categories: angry, disgust, fear, happy, neutral, sad, and surprise. Each image is 48 x 48 pixels.

The following chart shows the number of images per emotion for two datasets, *fer_ckplus_kdef* and *MMA Facial Expression*.



Since emotion *"contempt"* has only 130 images, the emotion has been dropped from the dataset. After dropping emotion *"contempt"*, class label *"disgust"* has the least number of images (5,337 images). Thus, a random sample of 5,337 images has been selected per emotion for model's development. The result dataset has 37,359 images with seven emotion labels: anger, disgust, fear, happiness, neutrality, sadness, and surprise. The dataset is then divided further into 60% train (22,542 images), 20% validation (7,472 images), and 20% test (7,472 images) using stratified random sampling to ensure all emotions in each dataset have an equal number of examples.

Project Design

Technologies

- Google Colab for training an emotion recognition model.
- Web framework: Flask
- Program language: Python
- Python modules: zipfile, numpy, pandas, sklearn, keras, tensorflow, livelossplot, matplotlib, seaborn, random, opencv

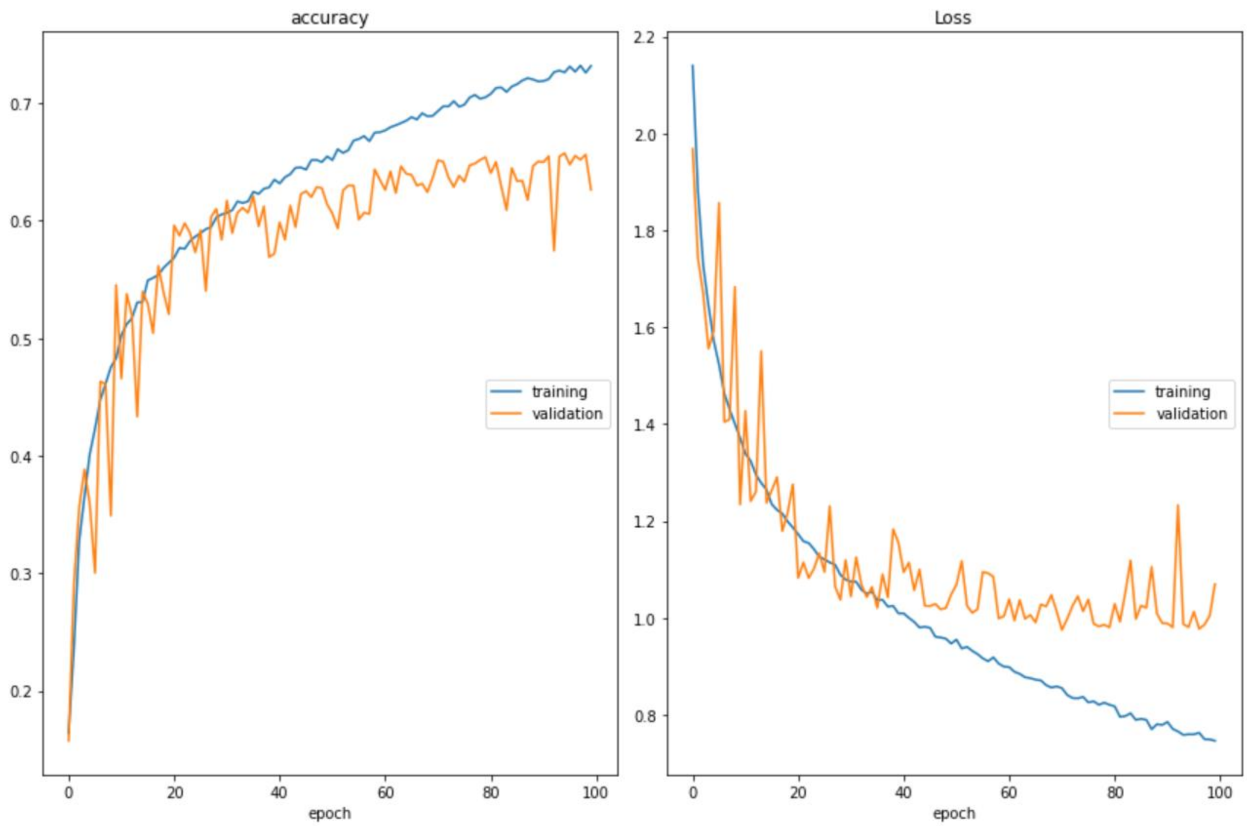
Model Development

The model was built and trained on a convolutional neural network. CNN architecture consists of four Conv 2D layers, each followed by a batch normalization layer, Relu layer, pooling layer, and dropout. In the end, we have two dense layers. The first Conv 2D layer contains 64 filters, Kernel size – (3, 3), input shape – (28, 28, 1), padding – same, and activated by a RELU activation function. The second Conv 2D layer contains 128 filters, Kernel size – (5, 5), padding - same, and activated by a RELU activation function. The third Conv 2D layer contains 512 filters, Kernel size – (3, 3), padding - same, and activated by a RELU activation function. The fourth Conv 2D layer contains 512 filters, Kernel size – (3, 3), padding - same, and activated by a RELU activation function. Pooling layer for each two-dimensional convolution layer will contain pool size – (3,3), strides – (2,2), padding – same. Then we have a flatten layer and two Dense layers where first dense layers contain 256 neurons, second dense layers contain 512 neurons. The two dense layers will be followed by a batch normalization layer, Relu layer, and dropout with value 0.4 Dropout. Finally, the output of the last dense layer is fed to a soft-max layer that assigns a probability for each class. The model was compiled using ADAM optimizer, categorical cross entropy function and learning rate of 0.001. We used callback with three functions that are ReduceLRONPlatreau, model check point and PlotLossesKerasTF. We used ReduceLRONPlatreau to reduce learning rate with no improvement in the metric with minimum learning rate – 0.0008, monitor – validation loss, factor – 0.1 and mode - auto. Model checkpoint was used to save the model weights at frequent intervals with monitor – validation accuracy, save weights only – True and mode - max. PlotLossesKerasTF method was used to create learning curve plots of loss, accuracy on training and validation set. Trained the network by using a fit method where we set the model with 100 epochs, batch size with 32, validation/training data and passing callback function.

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 28, 28, 64)	640
batch_normalization (BatchNo	(None, 28, 28, 64)	256
activation (Activation)	(None, 28, 28, 64)	0
max_pooling2d (MaxPooling2D)	(None, 14, 14, 64)	0
dropout (Dropout)	(None, 14, 14, 64)	0
conv2d_1 (Conv2D)	(None, 14, 14, 128)	204928
batch_normalization_1 (Batch	(None, 14, 14, 128)	512
activation_1 (Activation)	(None, 14, 14, 128)	0
max_pooling2d_1 (MaxPooling2	(None, 7, 7, 128)	0
dropout_1 (Dropout)	(None, 7, 7, 128)	0
conv2d_2 (Conv2D)	(None, 7, 7, 512)	590336
batch_normalization_2 (Batch	(None, 7, 7, 512)	2048
activation_2 (Activation)	(None, 7, 7, 512)	0
max_pooling2d_2 (MaxPooling2	(None, 3, 3, 512)	0
dropout_2 (Dropout)	(None, 3, 3, 512)	0
conv2d_3 (Conv2D)	(None, 3, 3, 512)	2359808
batch_normalization_3 (Batch	(None, 3, 3, 512)	2048
activation_3 (Activation)	(None, 3, 3, 512)	0
max_pooling2d_3 (MaxPooling2	(None, 1, 1, 512)	0
dropout_3 (Dropout)	(None, 1, 1, 512)	0
flatten (Flatten)	(None, 512)	0
dense (Dense)	(None, 256)	131328
batch_normalization_4 (Batch	(None, 256)	1024
activation_4 (Activation)	(None, 256)	0
dropout_4 (Dropout)	(None, 256)	0
dense_1 (Dense)	(None, 512)	131584
batch_normalization_5 (Batch	(None, 512)	2048
activation_5 (Activation)	(None, 512)	0
dropout_5 (Dropout)	(None, 512)	0
dense_2 (Dense)	(None, 7)	3591

Model Evaluation

The following figure shows the accuracy and loss for training and validation per number of epochs. The validation's accuracy seems to stay almost the same after 60 epochs. Similarly, the validation's loss does not change much after 60 epochs. Lastly, the maximum validation's accuracy is 65.8%.

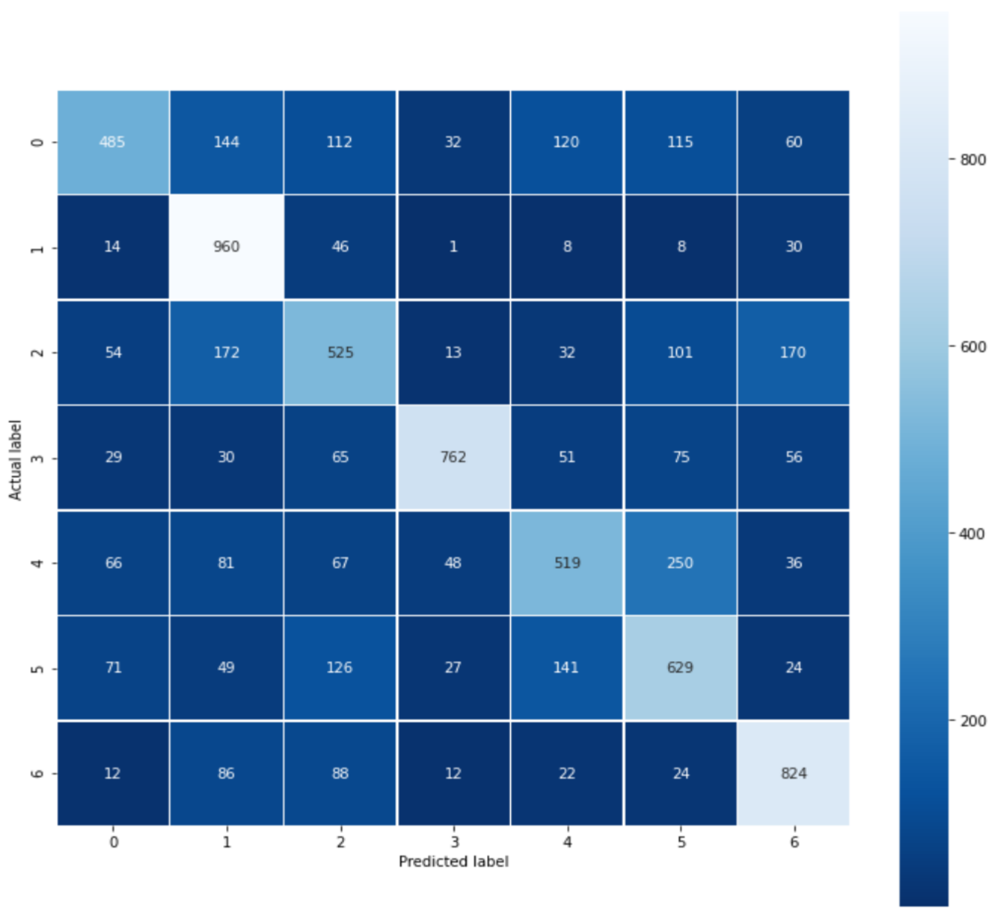



```
accuracy
        training             (min:    0.164, max:    0.732, cur:    0.732)
        validation           (min:    0.157, max:    0.658, cur:    0.626)
Loss
        training             (min:    0.746, max:    2.140, cur:    0.746)
        validation           (min:    0.975, max:    1.968, cur:    1.070)
```

The classification report below shows the scores for precision, recall, F1-score, and support for each emotion label in which emotion 0 to 6 indicates anger, disgust, fear, happiness , neutrality, sadness, and surprise, respectively. According to the report, emotion "*happiness*" (labeled as 3) has highest precision and F-1 score, and emotion "*disgust*" (labeled as 1) has highest recall. All emotions have roughly the same value for support. The accuracy on the test set is 63%, and the weighted average is almost the same for precision, recall, and F1-score.

	precision	recall	f1-score	support
0	0.66	0.45	0.54	1068
1	0.63	0.90	0.74	1067
2	0.51	0.49	0.50	1067
3	0.85	0.71	0.78	1068
4	0.58	0.49	0.53	1067
5	0.52	0.59	0.55	1067
6	0.69	0.77	0.73	1068
accuracy			0.63	7472
macro avg	0.64	0.63	0.62	7472
weighted avg	0.64	0.63	0.62	7472

The confusion matrix shown below indicates that emotions “disgust”, “*happiness*”, and “surprise” (labeled as 1, 3, and 6, respectively) have high true positives while emotions “*anger*”, “fear”, and “neutrality” (labeled as 0, 2, and 4, respectively) have low true positives.



Work Plan

The team communicated through Discord and created lists of tasks to schedule the work and weekly meetings every Thursday to discuss tasks and assign them. Collaboration was done through google drive, GitHub repository and Google Colab. The following weekly task are listed below:

- Reading the articles, exploring the datasets, and finalizing on a single dataset. Task was finished in 1 week. (Feb 9 – Feb15).
- Data Augmentation: filter/partition imbalance data, image transformation, preprocessing layers. Task was finished in 3 weeks. (Feb16 – Mar2).
- Model – setup environment, decide on the architecture, train the model, and evaluate the model. Task was finished in 5 weeks (Mar2 – Mar 30).
- Metrics – graphs, accuracy. Task was finished in 1 week (Apr1 – Apr7).
- Improve accuracy. Task was finished in 2 weeks (Apr8 – Apr15).
- Flask openCV image capture and perform classification for that image with a trained model. Task was finished in 2 weeks (Apr8 – Apr15).
- Project's report. Task was finished in 1 week (Apr16 – Apr 23).

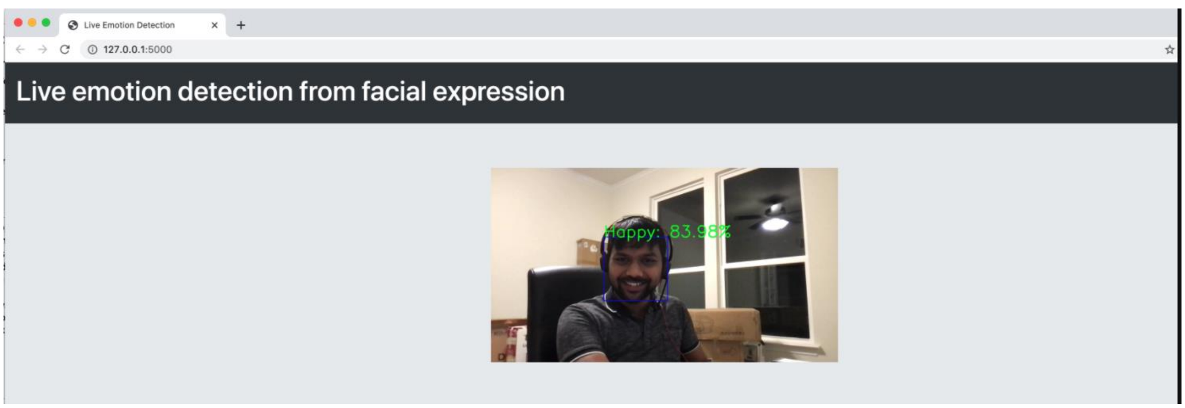
#	9-Feb	16-Feb	23-Feb	2-Mar	9-Mar	16-Mar	23-Mar	30-Mar	1-Apr	8-Apr	15-Apr	19-Apr
Description	W1	W2	W3	W4	W4	W5	W6	W7	W8	W9	W10	W11
Reading the articles, exploring the datasets and finalizing on a single dataset												
Data Augmentation												
Model - setup environment, decide on the architecture, train the model, and evaluate the model												
Metrics – graphs, accuracy												
Improve accuracy												
Flask-openCV-image capture and perform classification for that image with trained model												
Projects report												

Conclusion

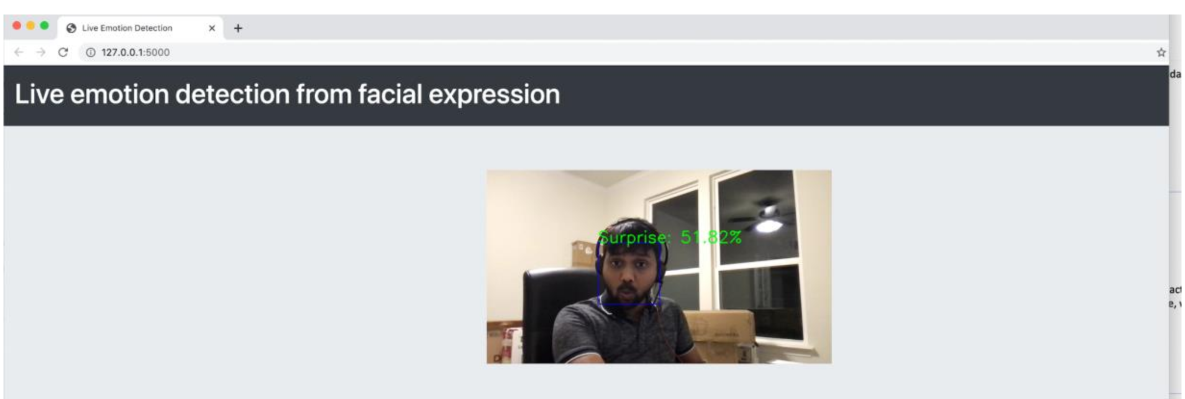
- Two datasets from kaggle were used for training our model.
- 6 Classified emotions - 'Angry', 'Disgust', 'Fear', 'Happy', 'Neutral', 'Sad', 'Surprise'.
- Model has been trained using CNN architecture using keras and other helper libraries.
- From the confusion matrix - the least accurate prediction class is ‘Angry’ and most accurate predictions are for ‘Disgust’ class.
- Website is built on Flask.
- openCV is used on the website to capture video feed from a webcam and detect the faces.
- Trained model is then integrated with detected faces to display the emotion and corresponding percentage of the detected emotion class.
- Website is even able to detect multiple faces in a given frame and perform emotion recognition.

Demo Screenshots from Website:

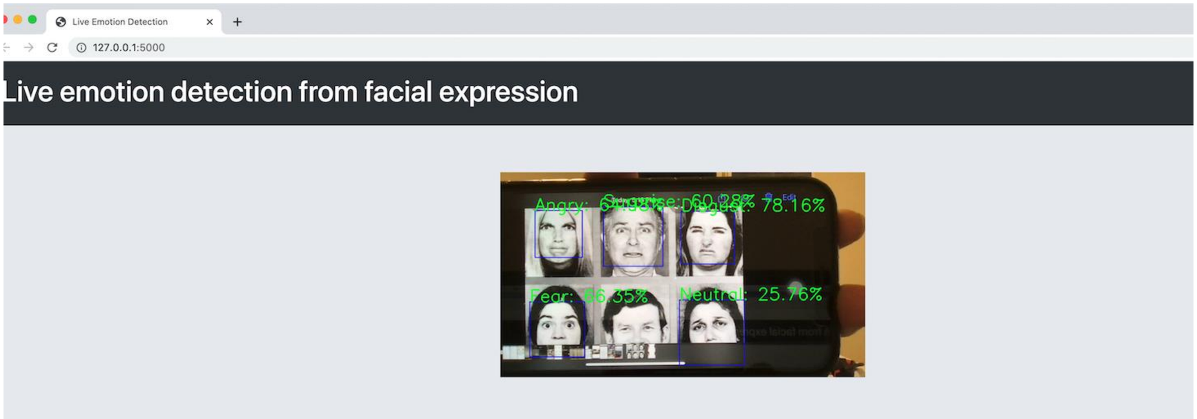
Happy Emotion Detection:



Surprise Emotion Detection:

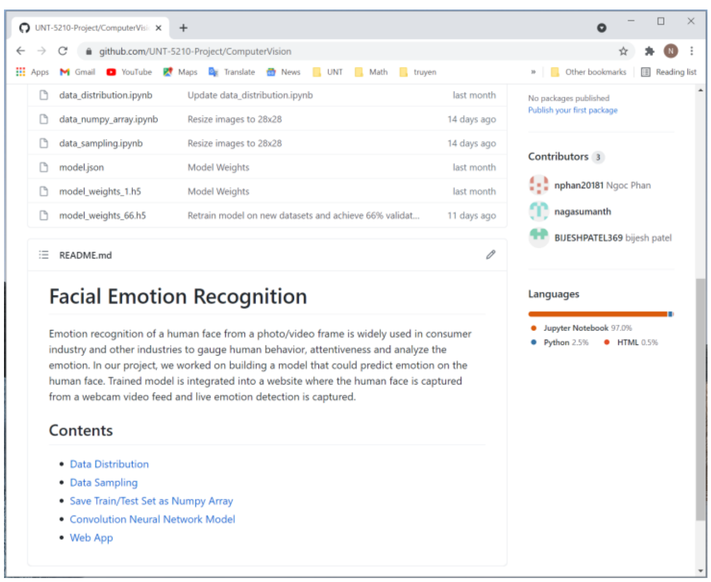


Multiple faces emotion detection:



GitHub Repository

- Url: https://github.com/UNT-5210-Project/ComputerVision
- Screenshot



References

- Flask - https://flask.palletsprojects.com/en/1.1.x/
- OpenCV - OpenCV - OpenCV, https://github.com/karansjc1/emotion-detection/tree/master/with%20flask
- Datasets - https://www.kaggle.com/sudarshanvaidya/corrective-reannotation-of-fer-ck-kdef, https://www.kaggle.com/mahmoudima/mma-facial-expression