Human Emotion Recognition in Real-time videos

Smita Subhadarshinee Mishra
Department of Computer Science
Golisano College of Computing and Information Sciences
Rochester Institute of Technology
Rochester, NY 14586
sm8528@cs.rit.edu

Thomas B. Kinsman,Ph.D
Department of Computer Science
Golisano College of Computing and Information Sciences
Rochester Institute of Technology
Rochester, NY 14586
thomask@rit.edu

***Abstract*—Deep Learning in Computer Vision is making new discoveries in object detection and classification, as well as image segmentation. At the same time facial feature analysis in videos and images has been a comparatively difficult task in machine learning. Facial recognition is useful for various applications and has been explored a lot in the recent years. Emotion detection is one of the many applications of facial recognition. Main objective of this paper is to sight a face in real-time and detect the region of interest to predict the emotion of the subject in the frame. We are putting the emotions into 7 buckets neutral, happy, angry, sad, surprised, disgust and fear. In order to make this happen we use a database with labelled data of pixel values that correspond to the Region of Interest(ROI) and use Keras to create the learning model which takes the ROI from real-time video input frames and predicts the possibility percentage for each emotion. The model gives us the probability that the subject in the input frame is demonstrating each of the 7 emotions and then we predict that the detected emotion is the one with maximum probability. This would enable us to detect the emotion and use if for various other applications.**

I. Introduction

Facial expression reflects the emotional state of a human while communicating and humans rely more on non-verbal communication. Though primarily verbal communication is used but a lot can be derived from a person's expression. For instance when human is shocked or surprised they are unable to express with words. Further when someone is lying, the it is easier to detect from their facial expressions rather than the words they use.

This feature can be put to use in various scenarios. Consider a scenario where a psychologist has to understand what the client might be feeling by analysing the body language and facial expression to diagnose better, since non-verbal communication reveals more about the client's emotion. Another scenario could be detecting drowsiness of a person driving. By detecting this we could provide alarm and stop the car to prevent accidents. As we know the world is progressing a lot towards robots and human labour is also being replaced by robots and this application of understanding human emotion would help them understand human needs better to serve them, especially in case of elderly people. We could also use this to check the effectiveness of a marketing campaign.

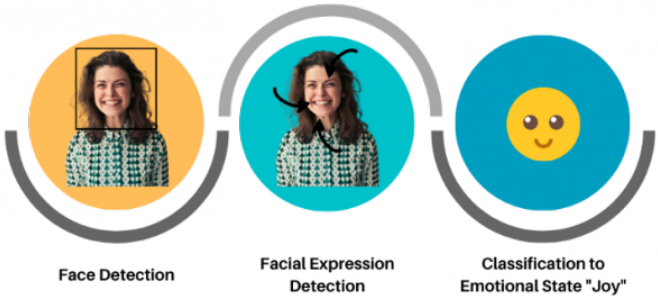

Fig. 1. For emotion detection we first detect the face of the subject in a frame and then take the Region of interest in the facial expression, finally we classify it into one of the know emotional states. [1]

This project aims on reading video inputs in real-time and predict the current emotion of the person in the frame. Paul Ekman found that facial expressions can be categorized into six buckets in 1970s (happy, sad, angry, fear, surprised and disgust) and is common to people from different cultures. However, people from different backgrounds might have some influence on their expression. So it is important to take all these constraints into consideration while designing the model. To achieve this the first step is getting a data-set and prepossessing it so that it can be used by a deep learning model. These prepossessed data should be then split into training and testing set. Then pass the training set into the deep learning model to understand the data and create a prediction model which I plan on using later, on the real-time video inputs to predict the emotion and put it into one of the 5 categories. To test our model I plan on using the testing set where we know the expected output and compare it with the output that we get to measure the accuracy.

II. Background

We all know the pain of understanding one's sentiment. The thousands of different emotions can be put into 6 categories happy, sad, angry, fear, surprised and disgust.

As per research on this area the current advancement on sentiment analysis is based on Natural Language Processing (NLP) where they gather various data from Twitter and Facebook posts and use that as a training set to predict the sentiment that is been conveyed in the future posts. [3] The

Fig. 2. The six Buckets of expression to predicted in this project(figure from TBD) [2]

major social media blogging sites like twitter, where people put their tweets/status message which could be written within 140 characters and could convey the emotions of the person. NLP helps us define and classify the sentiment polar opinions. Further NLP doesn't require the whole text to predict the sentiment, instead it just works on the important parts of the text to derive information regarding the sentiment to predict the subject's emotion.

Another paper [4] states how emotion can be predicted from videos.For machine learning models, analyzing faces in videos and photos has proven to be a challenging issue. Deep learning techniques have recently been employed, and the outcomes and reliability have significantly improved. These techniques can be used to solve issues like face identification, emotion recognition, and emotion reaction prediction. In the case of an emotion reaction, pertinent information about the emotions in each frame frequently needs to be combined with the speech signal and a succession of frames with varying lengths in order to make a meaningful forecast. A subset of the sequence analysis task known as emotion reaction prediction mainly relies on dynamic temporal and spectral variables. Convolution neural networks (CNNs) have been utilized widely and successfully for emotion recognition issues. They cannot, however, model an emotion transaction and cannot extract time-series data from a set of inputs. Recurrent neural networks (RNNs) are widely employed in the field of sequence analysis due to their capacity to produce outstanding outcomes on a number of tasks. We suggest a system for video emotion recognition and reaction prediction in this paper. The main focus is on an experimental investigation of a hybrid CNN-RNN architecture for emotion transaction analysis, which can identify an emotion in a video frame and forecast the right response.

Further there has been some work on emotion analysis from real-time video inputs using deep learning model [5]. A broad framework for generating Convolutional Neural Networks (CNNs) is suggested in one of the studies as a method for creating real-time CNNs. By using our suggested CNN architecture, they develop a real-time vision system that simultaneously performs the tasks of face detection, gender classification, and emotion classification. This serves as a validation of their models. They proceed to test against common benchmark sets after outlining the setup elements of the training approach. They claim that their accuracy in the FER-2013 emotion dataset (66 percent) and IMDB gender dataset (96 percent) respectively. Additionally, they debuted the brand-new guided back propagation visualization technique with real-time capabilities. They contend that in order to close the gap between slow performances and real-time architectures, careful application of contemporary CNN architectures, utilization of current regularization techniques, and display of hitherto hidden features are required.

In this paper we are going to talk about emotion prediction in Real-time data using Deep Learning model.

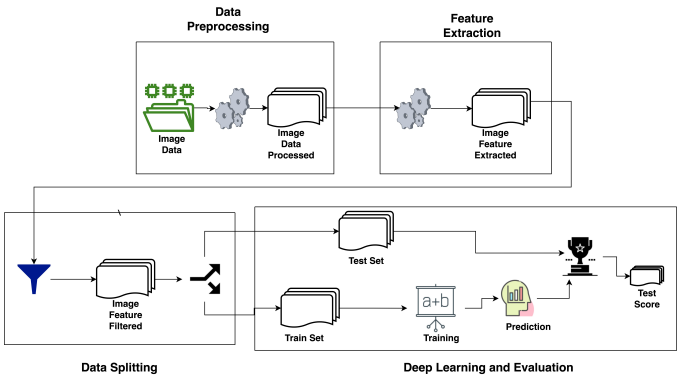

Fig. 3. Four phases of the project: A. Data Pre-processing: Converts the images into machine learning acceptable format. B. Feature Extraction: Extracts the important features in the image frame. C. Data Splitting: Split the data into training and testing sets. D. Deep learning and Evaluation: Fit the training data to predict and evaluate.

The project is divided into 4 phases as shown in Figure 3. The first phase is **Data processing** where we collect the data and use OpenCV to read and process the data to make is compatible for the Deep Learning Model. After that in the **Feature Extraction** phase we extract the important features in the image data and once we have that we filter out the information that is not required. Then the third phase **Data Splitting** phase we divide the data into Training and Testing set. Finally in the **Data Learning and Evaluation** phase the training set is passed through the Neural networks to fit it and create a model which is used for predictions and the test set is used to test the model and give the accuracy of the trained model.

III. Experimentation

To achieve my goal at first I tried creating my own data-set as I was not able to find a data-set that would work for my model. I tried creating **Optical flow** for the important points

on the face. Optical flow is a method to track the movement of pixel values in an image. There were two approaches to do this, **Feature point tracking** and **Dense flow tracking**. Feature flow tracking is a type of Sparse optical flow tracking where only the most important points are tracked, on the other hand Dense optical flow tracking takes into consideration all the points in a frame and shows how the pixels have moved over time. I started by collecting video samples from friends where they were demonstrating the different emotions. The code was reading these video frames and then pre-processing it using openCV to make it compatible for the learning model. Then I used HAAR Cascade to detect a face in the frame. **HAAR Cascade** follows **Viola Jones** face detection technique, It is a widely known algorithm that is used to detect objects, for instance face in an image or a real-time video frame. I used frontal face detector that was made using HAAR cascade. Then I created an **Optical flow** for each video, to do so I iterated over each frame and ran the opnCV2 optical flow methods. After doing this over a sufficient number of frames I saved this into the database and labelled it with the corresponding emotion. I tried using this to learn the model, however the because of a small database the model model accuracy was very low(26 percent) and the loss was high.

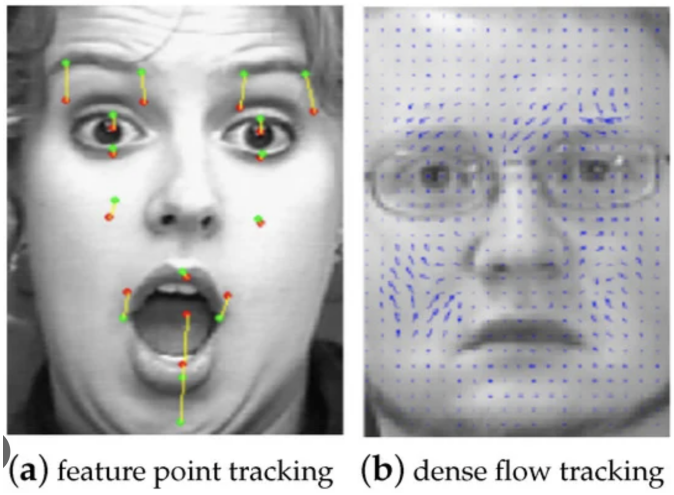

Fig. 4. The image (a) shows feature point tracking where the Region of interests in a face are tracked over series of frames. Image (b) shows the Dense Optical flow tracking of an image where all the pixels in an image are tracked over time. [6]

Later I found a database on kaggle [7] that was satisfying my requirements and used that to learn my model. The data-set that I found had pixel values corresponding to the ROI labelled with the corresponding emotion. I used this data-set and passed it into the CNN to create our learning model. Then I started reading real-time video input and pre-processing it by using **Opencv** to extract the region of interest and pass it into the model to get the prediction probabilities that the subject's present emotion belongs to each of the 7 buckets of emotions. To create the evaluation model I tried few approaches, first I tried using the model object that was used to fit the training set and there were runt-time errors because of the mismatch in the type of data the model expects. Later I tried loading the saved learning model from my files and passes the test set into the model to get the predictions. Finally used **sklearn** metrics to create the confusion matrix and get the accuracy score however while attempting to do that I got data mismatch error and hence used **Image data generator** from Keras to pre-process the test inputs and predicted outputs.

IV. Implementation

The first step that goes into process is collecting a large enough data-set that would go into the CNN.

A. Data Set

The data set used in this project was obtained fro, Kaggle [7]. The data consists of 48x48 pixel gray scale images of faces. The faces have been automatically registered so that each face roughly fills the same amount of space in each image and is roughly centered. The aim is to classify each face into one of seven categories based on the emotion displayed in the facial expression (0=Angry, 1=Disgust, 2=Fear, 3=Happy, 4=Sad, 5=Surprise, 6=Neutral). "Emotion" and "Pixels" are the two columns in the training set. The emotion that is present in the image is represented by a numeric code in the "emotion" column that ranges from 0 to 6, inclusive. For each image, a string enclosed in quotes is present in the "pixels" column. The values in this string are separated by spaces and are arranged in row major order. The single column in the testing set is "pixels," and our objective is to make predictions for emotion column. The training set consists of 28,709 examples. The test set consists of 3,589 examples.

B. Keras

Keras is built on top of **TensorFlow**. It is an open source end-to-end Application Programming Interface(API) for Python [8]. It is used to build Machine Learning models and makes it easier to create a Neural Network. Here we are using Functional API with a Single input and a single output using **Softmax**. In functional API we describe the Input layer by specifying the Shape of the data.

C. Framework

The following steps are followed while working on the video to predict the emotion.

1) Collect and process video inputs: To pre-process the image we first convert the image to Gray scale and use HAAR cascade [9] to detect face. HAAR Cascade is a widely used algorithm to detect face and other facial features, It uses Viola Jones algorithm in the background to achieve this. We are using a Haar cascade that would detect frontal face, Hence when we turn our face to different angles then the code would fail to detect the face. Once we have the detected region we select the detect face and extract the region of interest(ROI) from the image. ROI gives us the important features in the image that would be helpful in tracking the change in the facial muscles as the expression changes. The we convert the

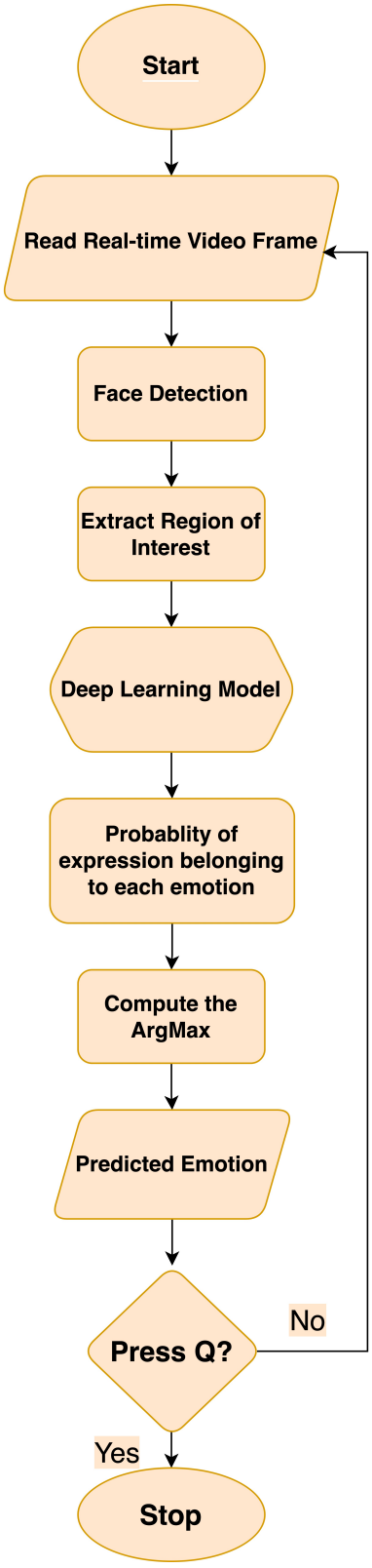

Fig. 5. The flowchart describes the flow of the project framework from reading the data-set and creating the model, finally predicting the emotion of real-time video input.

ROI to array. This would match our data to the data-set that we use for model training.

2) Feature extraction: Region of interest(ROI) gives us the important features in the image that could be useful for prediction. ROI is the sub region within a frame where our object of interest is present is called. We convert this ROI into an array to match with the model input requirements before passing it into the model.

3) Data Splitting: The data-set is split by assigning 70 percent for training and 30 percent of the data goes into testing. **sklearn** is used to split the data into training and testing set. The randomly selected 70 percent data is used for training the model and is passed into the Neural network. Rest of the 30 percent of data is used to evaluate the model accuracy.

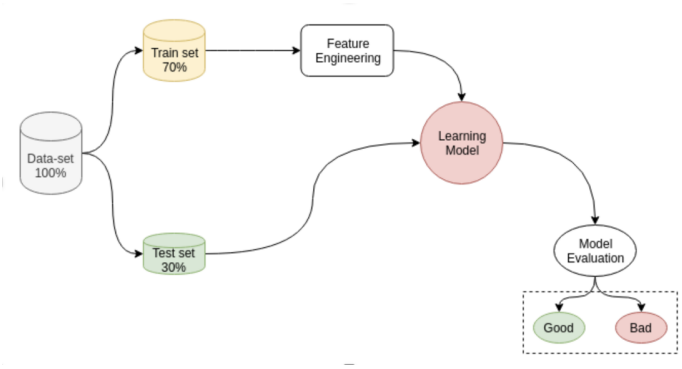

Fig. 6. Data splitting: Data is split into training(70 percent) and testing(30 percent) sets [10]

4) Deep learning machine: We are using **keras** to create the Neural Network using Functional API.The parameters for the models are chosen as per the model accuracy and what worked best for the current model has 10000 Epochs with batch size 32 to classify the output into 7 classes. The Neural Network has 4 convolution layers, each layer consists of a 2D Convolution layer, ReLu activation function, Batch Normalization and MaxPooling. **ReLu** activation function prevents the exponential growth when the model is scaled, **Batch Normalization** reduces the internal covariant shift hence accelerating the training process and Finally **MaxPooling** is used to reduce the dimension of the images.

5) Check accuracy using the testing dataset: The labeled data-set is split into training and testing set. The test set consists of 30 percent of the labelled dataset and is used against the Deep Learning model to calculate the accuracy of the model. We are using **sklearn** to create a Confusion Matrix that would help us distinguish between the correct prediction and the incorrect one. In the figure we can see the confusion matrix and the values diagonal from top left to bottom right are the counts for correct predictions.

V. Results and Discussion

The program reads the real-time input and passes it into the Deep Learning model to get the prediction. The model output is an array of length 7 with probability values corresponding

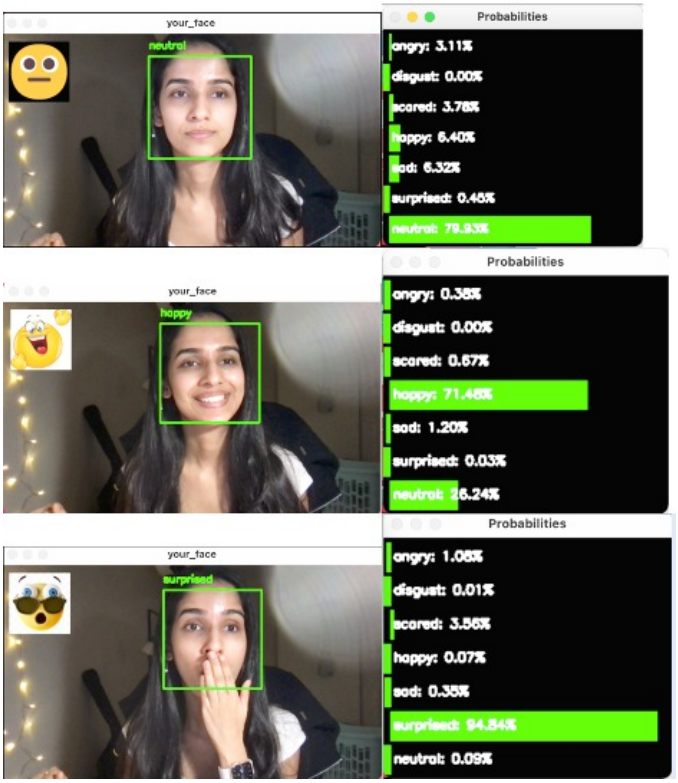

Fig. 7. The windows on the left show the real-time output of the detected face and emotion along with an emoji corresponding to the emotion. The right side windows show the probabilities that the frame in the left window belongs to each of the emotion bucket.

to the 7 emotions **[”angry” ,”disgust”,”scared”, ”happy”, ”sad”, ”surprised”, ”neutral”]** This indicates the array placeholders for each emotion and an example output of the model looks like [0.064 0.000 0.024 0.102 0.055 0.024 0.727]. Each probability value indicates the possibility of the emotion of our subject belonging to the emotion at the placeholder array position. Once we have this output, we are taking the **argmax** to get the array position corresponding to the maximum value in the output array and finally we select and display the corresponding emotion. The output of the application is real-time video frames showing the detected face and emotion. There is another output window that displays the probabilities of the 7 emotions.

The program displays the real-time video input as we see in fig 7 with the detected face highlighted with a green rectangle and the corresponding emotion labelled above the rectangle. There is an emoji that says which emotion we are talking about in the top left corner of the frame. The figure 7 shows us 3 different frames of this output along with a window that displays the model output, which is probabilities of the person’s emotion belonging to each of the bucket.

We are creating a Confusion Matrix from the actual and predicted values and plotting it using NumPy. Confusion Matrix is a visual representation of the number of correct and wrong predictions that helps us asses the errors in our model. In the figure we can see the confusion matrix and the values

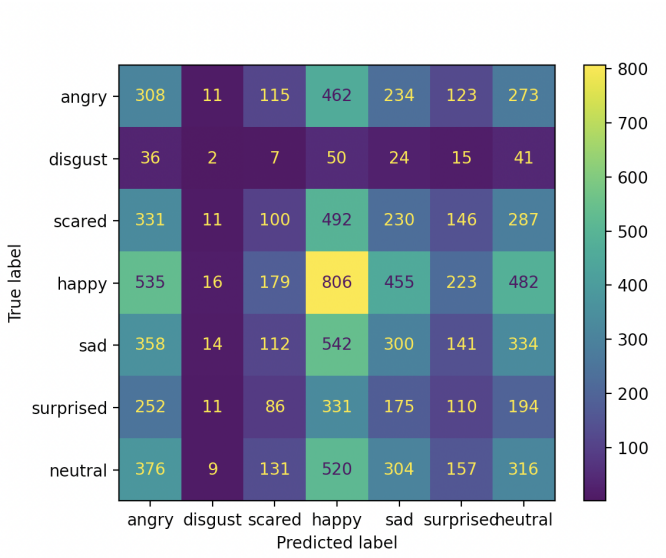

Fig. 8. The confusion matrix helps us visualize the number of correct predictions and false predictions. The values diagonal from top left to bottom right are the counts for correct predictions.

diagonal from top left to bottom right are the counts for correct predictions.

VI. Future Work

While having a conversation we see the facial expression of a person and combine it with what they say to conclude their emotional reaction. Our model would predict more accurately if we take voice inputs into consideration. We could create another learning model that takes voice input and gives a prediction for the emotion, finally we could compare the percentages of both the model outputs to decide on the emotion selection. This would mostly increase the accuracy by avoiding the false positives and increasing the true positives for disgust (Fig 8) and other emotions. Further we are using HAAR cascade here to detect Frontal face, hence we could consider detecting face from different angles. I feel we could add more diverse data into our data set along with data for side face detection. Finally using Optical flow to detect the motion in the different ROI points in the detected face would help us stay away from the bias caused by the influence of region on the emotion.

VII. Conclusion

Shankar Vedanta stated ”Social study research say that our emotional tie to others, shape our well-being” [11] in one of his Pod castes from Hidden Brain. The emotional state could even affects our physical health. He continues to talk about the importance of something as simple as a Smile to a stranger. Studies have proved that we are affected by the emotional state of people around us hence it is very important to understand ones feelings and act accordingly. As we are moving towards a more computer dominated world, humans will be more and more dependent on machines to get their work done. Now this

increases the need for machines to understand human emotions to get serve human needs better.

The paper describes how to build an emotion detection model using Computer Vision and Deep Learning concepts. This model takes real-time video inputs and predicts the emotion of the person in the frame. We have a labelled dataset that is spilt into test and training set which is used to train and evaluate the model respectively. Keras is used to train the model, which internally creates a Neural Network with 4 layers to fit the training set. The real-time video input is pre-processed using OpenCV to extract the Region of Interest and make it suitable for the Deep Learning Model.

The output from the model is probabilities of the detected emotion belonging to each of the 7 buckets of emotion. Then we take the emotion with maximum probability. The model output is displayed with the real-time video input where the detected face is highlighted and their is an emoji corresponding to the emotion on the video frame on the top left. There is another window that shows the probabilities corresponding to different emotions.

ACKNOWLEDGMENT

I am glad to have this opportunity to work on this project to detect emotion from real-time videos, which allowed me to dive deep into Neural Networks and Image processing. I would like to thank my Advisor Dr Thomas B Kinsman for guiding me through the whole process and the project. I am glad I got the opportunity to work with him and he has mentored me for the majority of my time in Rochester Institute of Technology(RIT). My courses related to Computer Vision that I took under the guidance of Dr Kinsman encouraged and motivated me to come up with this project idea and implement it.

I would like to thank my Instructor Professor Carlos R Rivero who made it easier to understand the different milestones and help deliver my work on time. He structured the capstone timeline in a way that I could divide my work into different milestones and simultaneously work on my report and poster. Finally I would like to extend my appreciation towards Professor Hans-Peter Bischof and the Computer science department for the help with the materials and knowledge that helped me deliver this project successfully.

REFERENCES

[1] K. VEMOU and A. HORVATH, "Facial emotion recognition." [Online]. Available: https://edps.europa.eu/data-protection/our-work/publications/techdispatch/techdispatch-12021-facial-emotion-recognition_en

[2] Z. Jamaludin, "Basic emotions." [Online]. Available: https://www.researchgate.net/figure/Six-basic-emotions-Source-https-managementmaniacom-en-six-basic-emotions_fig2_312160510

[3] B. Selvaretnam, W. Y. Chong, and L.-K. Soon, "Natural language processing for sentiment analysis: An exploratory analysis on tweets." Dec. 2015. [Online]. Available: https://ieeexplore-ieee-org.ezproxy.rit.edu/document/7351837

[4] N. Ronghe, S. Nakashe, A. Pawar, and S. Bobde, "Emotion recognition and reachtion prediction in videos," Dec. 2020. [Online]. Available: https://ieeexplore-ieee-org.ezproxy.rit.edu/document/8234476

[5] O. Arriaga, P. G. Ploger, and M. Valdenegro, "Real-time convolutional neural networks for emotion and gener classification," Oct. 2017. [Online]. Available: https://github.com/oarriaga/face_classification/blob/master/report.pdf

[6] Y. Huang, F. Chen, S. Lv, and X. Wang, "Facial expression recognition: A survey," 2019. [Online]. Available: https://www.mdpi.com/2073-8994/11/10/1189

[7] C. Tang, "Challenges in representation learning: Facial expression recognition challenge," Apr. 2013. [Online]. Available: https://www.kaggle.com/competitions/challenges-in-representation-learning-facial-expression-recognition-challenge/data

[8] B. Copeland, "What is a keras model," Jul. 2022. [Online]. Available: https://www.activestate.com/resources/quick-reads/what-is-a-keras-model/#:~:text=Keras%20is%20a%20neural%20network,built%20for%20you%20by%20TensorFlow.

[9] A. Alekhin, "Haar cascades." Apr. 2020. [Online]. Available: https://github.com/opencv/opencv/tree/master/data/haarcascades

[10] A. Bose, "Cross validation." [Online]. Available: https://towardsdatascience.com/cross-validation-430d9a5fee220

[11] S. Vedanta, "Relationships 2.0: The power of tiny interactions." [Online]. Available: https://podcasts.google.com/feed/aHR0cHM6Ly9mZWVkcy5zaW1wbGVjYXN0LmNvbS9rd1djMGxoZg/episode/ZDk0ZjY1MzUtNzA4Ny00N2FmLWI1ZjQtYTRiMTViMzUxYmFk?ep=14