Sprint 1 Deliverable

Visual-Based Detectors Tool-Kits:

1) Tensorflow, Keras, and OpenCV
    a) Utilizing Python and a combination of OpenCV and Deep Learning courtesy of Keras and TensorFlow, this solution enables detection of seven different emotions. Based on the comments provided by the developer, it appears to be capable of detecting only one face. However, when viewing the code there appears to be potential of modifying the code to be able to view multiple faces.
    b) The code utilizes python and libraries that are publicly available. Certain group members have previous experience with utilizing OpenCV. Provides the opportunity for students to gain experience with Deep Learning.
    c) TensorFlow is only supported on Python 3.5-3.7 which no longer receives support from Python. This may or may not cause future problems with other libraries.The effectiveness of this model is unknown and would need to be further researched. However, the simplicity of it would potentially make it good to emulate.
    d) https://python.plainenglish.io/real-time-emotion-detection-from-webcam-using-deep-learning-and-opencv-952953dbf051
    e) https://github.com/rajdeepUWE/Real-time-Emotion-Detection-from-Webcam-using-Deep-Learning-and-OpenCV
2) **Deepface**
    a) https://github.com/serengil/deepface
    b) Deepface is a lightweight face recognition and attribute analysis framework. It can hypothesize age, gender, emotion, and race. It supports a multitude of different state-of-the-art face recognition models. FaceNet, VGG-Face, ArcFace, and Dlib are its top models with more than a 99% accuracy, which surpasses human beings (97.53% accuracy).
    c) With more than two million downloads and more than 9.3k stars on github, deepface is a very popular toolkit for our purposes.
    d) Deepface handles all the common stages of face recognition in the background, so the barrier of entry is extremely low.
    e) The pre-trained emotion detector model provided by the deepface can label angry, disgust, fear, happy, sad, surprise, neutral. At first, we will probably limit the labeling to angry, neutral, happy.
    f) Since our project must take into account gauging the aggregate emotion of a room over the span of multiple people, some pre-processing must first take place to pass faces individually into the model.

g) Could very easily use multiprocessing/multithreading to speed up the individual emotional analysis of each face and then combine to determine the aggregate emotion

3) **PyTorch and FER - Facial Expression Recognition - Library**

a) This python library is structured to detect emotion from images, but can be used to detect emotion from video footage as well (via application of the model to several frames of a video stream)

b) At a high level, the steps to be able to detect emotion from video footage include:

i) Extracting frames at certain frame rate

ii) Preprocessing the frames to be in a format compatible with the PyTorch FER model

iii) Running inference on each pre-processed frame to obtain emotion predictions for each frame

iv) Aggregating results of all the inferences made

v) Displaying/outputting the result/emotional state

c) Typically PyTorch focuses on detecting emotions of a single person, but it can be adapted to handle the emotion detection of multiple people

i) This would require the use of a face detection algorithm prior to the emotion detection algorithm like Haar cascades, HOG + SVM, or deep learning-based detectors like MTCNN or dlib

ii) After this we would need to process each detected face/person, and aggregate results from the emotion detection library

d) The FER library typically detects the following emotions: Angry, Disgust, Fear, Happy, Sad, Surprise

e) Link on implementation/usage details (basic details listed here): https://pypi.org/project/fer-pytorch/

i) “Training is done using the synergy of Pytorch Lightning and Hydra packages for setting training loops and configs correspondingly”

4) **Facial Detection Model using CNN(Convolutional Neural Network)**

a) Advantages of CNN

i) Learn features from raw data, even if the object is not centered

ii) Models can be transferred for fine-tuning

b) Disadvantages of CNN

i) Requires a large dataset for training

ii) Sensitive to change in the input data (not robust)

c) Example using CNN https://www.analyticsvidhya.com/blog/2021/11/facial-emotion-detection-using-cnn/

i) Accuracy could be improved by using VGG-16 or Resnet, or using Stacked model

5) **Training Emotion Recognition Model using VGG model and FER dataset**

(The solution used to explore this approach uses VGG19 pre-trained on ImageNet.
Article Link: https://dagshub.com/blog/train-emotion-recognition-model/

repository: https://dagshub.com/GauravMohan1/Emotion-Classification)

a) VGG models excel on the ImageNet benchmark, showcasing strong performance and generalization skills in visual recognition tasks.
    i) They outperform other models like RestNet in generalization which is important in facial recognition.
    ii) They boast high accuracy in image recognition, employing deep networks and compact 3x3 convolutional filters.

b) The solution suggests two ways to handle imbalanced data sets, one way is to use ImageDataGenerator and the other is to pass class weights while the model is training. The second method showed better performance.

c) When using an imbalanced dataset, a lack of training samples for specific class labels could lead to inaccuracies.

<u>Response Generator Tool-Kits (Audio and/or Text):</u>

1) **OpenAI API**
    a) https://pypi.org/project/openai/
    b) Due to its advanced training model set, GPT-4 has been seen to have great emotional intelligence in its ability to understand and express emotions. As such, we can use prompt engineering to design GPT-4 as a response generator that will take into account the emotion of the room to modify an initial statement to better suit the environment.
    c) GPT-4 is an LLM that is trained on over 220 billion parameters, so the model is leaps and bounds ahead of the competition to generate accurate outputs.
    d) We can utilize OpenAi's Python API endpoint to connect and query the model. The drawback to using an API is that it requires an internet connection to connect to OpenAI’s servers and may have a delay depending on various factors like network speed.
    e) There exists no free option, so we will have to look at paying for the use of the API and models. GPT 3.5 is about 20 times cheaper than GPT 4, though 3.5 is a less powerful and capable model. Pay is per query.
    f) OpenAI also has a text-to-speech feature that can utilize one of the six possible AI generated voices to read the text. While it is currently not directly possible to change the range of emotion of the voice, it did state that it could be altered depending on the grammar and capitalization used in the statement.
    g) Documentation and tutorials is very detailed so development should be fairly straight forward

2) **Tensorflow TTS (Audio)**
    a) Opensource
    b) Utilize data provided by the real time emotion detector to create a state machine that will output a specific rate, tone, and gender that is best suited for that emotions state.

    c) If Tensorflow is the chosen emotion detector, it provides consistency to use libraries (i.e., both the TTS and emotion detector solution would be able to work with Python 3.7).
    d) There are many examples of how to utilize Tensorflow TTS available online.

3) **TextGPT, Pytorch (text)**
    a) This is a python library built on Open AI's GPT-3
    b) Would need a dataset to train models for generating text based on an input emotion (specific to airport announcement context)
    c) Then a neural network architecture suitable for text generation needs to be defined (ex. RNN - recurrent neural networks for text generation) - would need to ensure there is a way to pass in an input emotion here as well
    d) Then the model would need to be trained to generate text corresponding to the input emotion provided
    e) Video setup: https://towardsdatascience.com/text-generation-with-python-and-gpt-2-1fecbff1635b
    f) GPT-2 simple package python (details usage and how to pip install this package): https://pypi.org/project/gpt-2-simple/

4) **Bidirectional LSTM(Long-Short Term Memory) and Word2Vec (text)**
    a) Captures both past and future sequences and process in both forward and backward direction
        i) However, the complexity increases
    b) Capture long-term dependencies and contextual information, these features are helpful when predicting emotions in text
    c) Requires large dataset for training
    d) Example using BiLSTM and Word2Vec for emotion detection in text https://www.analyticsvidhya.com/blog/2021/10/emotion-detection-using-bidirectional-lstm-and-word2vec/

5) **Google Gemini**
    a) Google Gemini is a large language model similar to OpenAI's GPT-4. This new model surpasses previous conventional AI models and is able to recognize and respond to emotions.
    b) Since it is a large language model, the AI operates as a very sophisticated neural network trained on an extensive database.
    c) Gemini compares similarly to ChatGPT and its API use is free for up to sixty queries a minute.
    d) Being developed by Google and with developer having free access to the AI models mean that it will constantly improve and become more powerful
    e) We connect to Gemini through Google's API, which means we need an API key and an internet connection. Delays should be expected and will be variable depending on internet speed and number of requests for Google to handle at the given moment

6) **Scikit-learn or pandas and Pydub (Audio)**
    https://www.geeksforgeeks.org/working-with-wav-files-in-python-using-pydub/

Or Google Cloud Text-to-speech API (Audio)
https://codelabs.developers.google.com/codelabs/cloud-text-speech-python3#2

a) PyDub is a python library for audio manipulation, including playback with different tones and speeds.
    i) Might or might not help put emotion into the voice. (needs more research)
b) The Google Cloud Text-to-Speech API converts text into audio formats such as WAV, MP3, or Ogg Opus. It also supports Speech Synthesis Markup Language (SSML) inputs to specify pauses, numbers, date and time formatting, and other pronunciation instructions.
c) Sckikit-learn or pandas are machine learning libraries written for python that could help with announcement recommendations based on emotion detection.

Conclusion

For the emotion recognition portion of this project, we decided to use deepface for the emotional analysis of each face in an environment coupled with either OpenCV or FER to identify the faces that will be plugged into the deepface module. We chose these because FER bundles both emotional analysis and facial detection, while also having the capability of utilziing a MTCNN facial model which is better at detecting faces than OpenCV's Haar Cascade Model. However, OpenCV will be our backup in the case the FER is tedious to integrate with the rest of the project.

Likewise, for the response generation, we will utilize Google Gemini to manipulate a given announcement statement depending on the prevalent emotion of a room. We chose Gemini because it is free to use for developers, which OpenAI's API endpoint is strictly paid for. Further, it uses similar architecture to ChatGPT-4 so it should function extremely well in its use as a LLM. Lastly, its documentation is straightforward and being developed by a prominent company like Google means that it will consistently improve and become more accurate/powerful.