Emotion Recognition by Audio and Video

AI Lab: Computer Vision and NLP

Armando Coppola (2003964)
Francesco D'aprile (2016953)
Romina Gryka (1993783)
Sara Lazzaroni (1983548)

Academic Year 2023/2024

Contents

1 Introduction

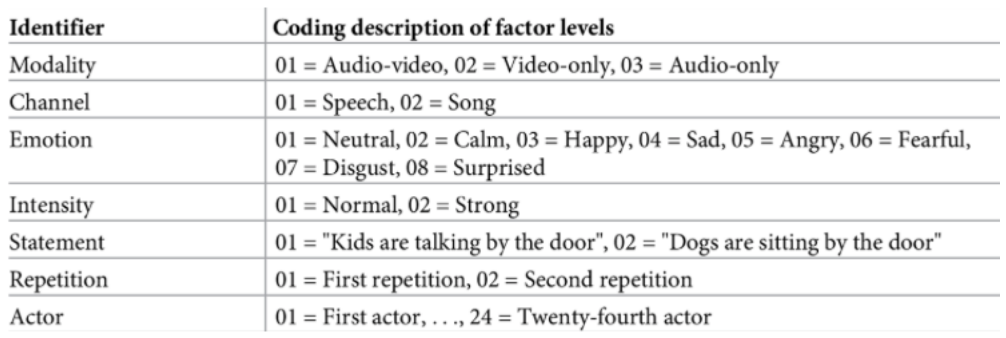
The application is designed to recognize human emotions through videos and audio. The project can have an important impact as emotions are the basis of social interactions. For example, from the point of view of human-machine interaction, having an AI model that recognizes the emotions of its interlocutor would allow the user experience to be personalized.
Another application could be the medical field. Indeed, the model could be used to monitor patients' state of mind and apply personalized medical therapies or it could be used to support people affected by Autism spectrum disorder (ASD), to deal with the difficulty of recognizing the emotions of people they interact with.
Furthermore, the software could be used to prevent violence and abuse by detecting emotions such as fear and anger.

2 Dataset used

Two different kaggle datasets were used to implement the entire project: RAVDESS dataset and RYERSON dataset.

2.1 RAVDESS dataset

The RAVDESS dataset is composed by 7356 recordings in English made by 24 professional actors, combining singing and voice. Recordings are divided into three formats: audio only, video only, video with audio. Each recordings is classified in eight emotions: neutral, calm, happy, sad, angry, fearful, disgust and surprised.
Various information are contained into the name of each file that is composed by 7-part numerical identifier as following:



Identifier	Coding description of factor levels
Modality	01 = Audio-video, 02 = Video-only, 03 = Audio-only
Channel	01 = Speech, 02 = Song
Emotion	01 = Neutral, 02 = Calm, 03 = Happy, 04 = Sad, 05 = Angry, 06 = Fearful, 07 = Disgust, 08 = Surprised
Intensity	01 = Normal, 02 = Strong
Statement	01 = "Kids are talking by the door", 02 = "Dogs are sitting by the door"
Repetition	01 = First repetition, 02 = Second repetition
Actor	01 = First actor, . . ., 24 = Twenty-fourth actor

The application only considers 1172 recordings of the audio-video format classified into six emotions (happy, sad, angry, fearful, disgust and surprised).

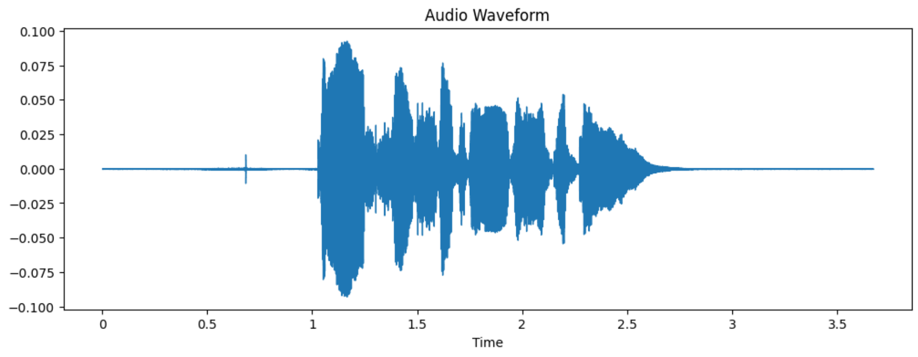
2.2 RYERSON dataset

The RYERSON dataset is composed by 728 recordings in six different lenguages (English, Mandarin, Urdu, Punjabi, Persian, and Italian). All recordings is in video-audio format. Each recordings is classified in six emotions: happiness (ha), sadness (sa), anger (an), fear (fe), surprise (su) and disgust (di). The filename of each file contains the corresponding label class.

3 Implementation

3.1 Audio Model

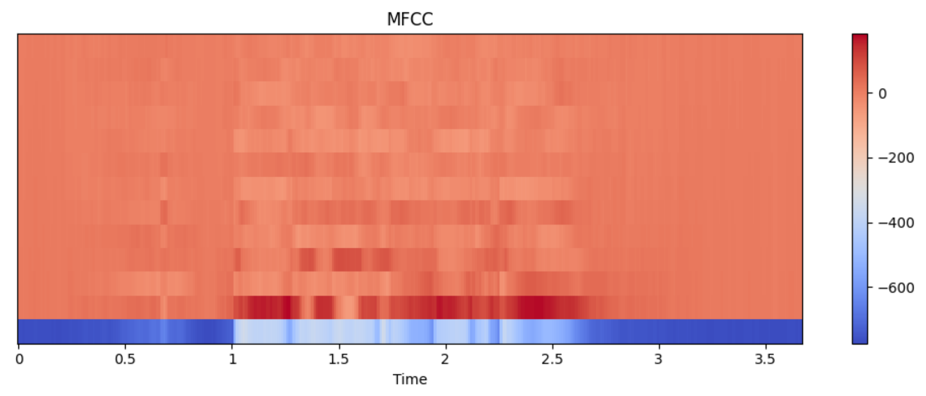
The Audio Model is used to recognize human emotions through audios.
The first step to take is to obtain the correct dataset to use, so the application starts with the extraction of audios from a list of video files, in order to have a directory in which the extracted audio files will be saved.



To reach the final goal there are different points to follow, these are listed and explained below.

Preprocessing of the dataset
The dataset is preprocessed as follow:
1. Iteration on audio file
2. Extraction of the six emotions from the file name (differently for ryerson and ravdess)
3. Extraction of MFCC features
4. Saving features in *.npy* files
5. Adding the path of the *.npy* file to the list

After obtaining the **dataset** and **dataloader** there's the model.



Model

For the creation of the model was used convolutional neural network (CNN), developed with Pythorch library, to extract local features from the spectrograms, pooling layers to reduce dimensionality, and fully connected layers to combine the extracted features and produce the final classification probabilities. This approach allows the model to learn complex patterns in audio data, improving classification accuracy.
Functioning of the model:
Input: batch of audio spectrograms
Convolutions and pooling: the spectrograms pass through four convolutional layers, each followed by a ReLU activation function and max pooling, progressively reducing the spatial dimension and increasing the depth of the feature maps
Flattening: feature maps are transformed into a 2D vector
Fully connected layers and dropouts: the vector passes through a series of fully connected layers, with dropouts to prevent overfitting
Output: the output is a vector of probability obtained through softmax

Train Loop

The train loop is composed by three function called *train audio model*, *create audio model*, *audio predict.*

- *train audio model(subtrain_dataloader, subtest_dataloader)*:
  This function trains a CNN model for the classification of audio spectrograms and is composed by the follow phases:
  1. Initialization of the model and confusion matrix to plot the predictions
  2. Setting the loss and optimizer
  3. Training loop, that iterates over the batches of the *subtrain dataloader*, the audios pass through the model to obtain the predictions, is calculated the loss between the predictions and the real labels,and the total loss is accumulated for the epoch
  4. Best model selection and saving, where is determined the best accuracy across all epochs, the state of the model corresponding to the best accuracy is retrieved

and the best model is saved in a file
5. The validation of the results, where there's the graph for the training loss for epoch, the graph of the validation accuracy for epoch and the confusion matrix to evaluate the performance of the model on the different classes of labels

- *create_audio_model():* This feature implements a complete flow for training the emotion recognition model from audio.
  The process includes the following phases:
  1. Extracting audio files from videos of the subtrain set and subtest set
  2. Extraction of audio features
  3. Extraction of feature labels
  4. Creation of the dataset and data loader
  5. Train the model using the *train audio model function*

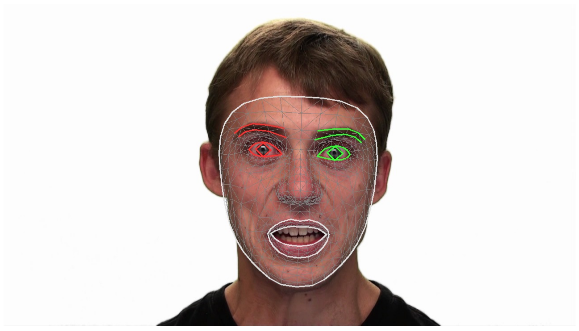
- *audio_predict(subtest_dataloader_audio):* This function predicts emotions using a previously trained convolutional neural network model. The process implements the following phases:
  1. Extracting Audio from Test Set Videos
  2. Extraction of audio features from previously extracted audio files
  3. Extraction of labels
  4. Creation of the Dataset and DataLoader
  5. Loading the pre-trained model
  6. Prediction on the subtest set
  7. Confusion matrix of the model on the subtest set
  8. Prediction on the test set 9. Confusion matrix of the model on the test set
  10. Returning a matrix of probabilities of the following shape: [6 emotions, number of predictions] for subtest set and test set

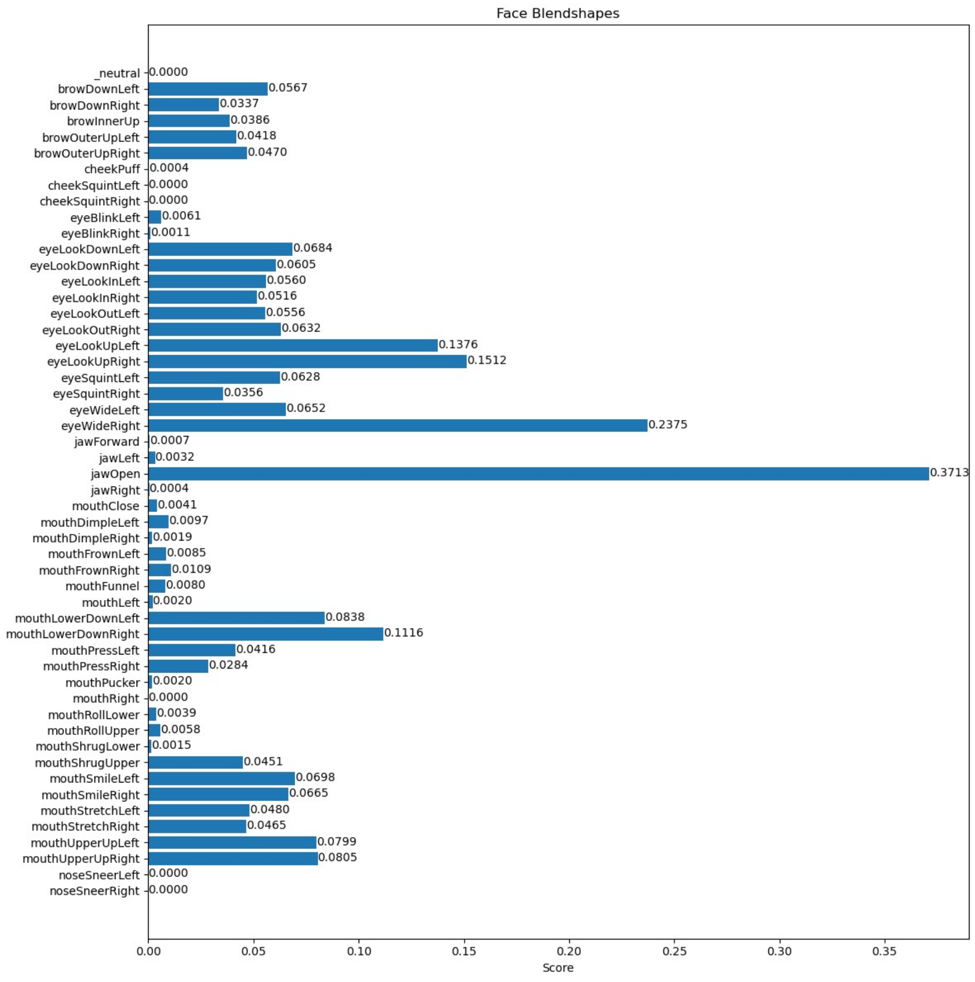
3.2 Video Model

The Video Model is used to recognize human emotions analyzing videos.
A list of video files is obtained starting from the datasets mentioned previously.
Then, to predict emotions from this video, various steps are followed that are explained below.

Preprocessing of the dataset
The dataset is processed as follow:
1. Loading of each video
2. For each video, extraction of 50 frames
3. For each frame of video, extraction of facial key-points from which 51 values are

obtained that represent particular facial features such as jawOpen or eyeWideRight. These are the features extracted for each frame. In this way, we obtain the list of lists that represents the change in facial features during the entire video





4. Saving features of all videos into a *.pkl* file
5. Extraction of emotion labels from the name of the video files (differently for ryerson and ravdess)

Model
As the model, a Long Short-Term Memory (LSTM) network, developed with the TensorFlow library, is used to capture temporal dependencies in video data. This approach allows the model to learn complex patterns over time, improving the recognition of emotions from video sequences. Dropout layers are utilized to prevent overfitting, and fully connected layers combine the extracted features to produce the final classification probabilities.
Functioning of the model:
Input: features extracted from each video with a specified shape
LSTM Layers: the video sequences pass through two LSTM layers. The first LSTM layer returns sequences, which are then passed to the second LSTM layer that processes these sequences. Both LSTM layers have dropout to avoid overfitting
Fully Connected Layers: the output from the LSTM layers is passed through a fully connected layer with ReLU activation function
Output: the final dense layer with softmax activation produces a vector of probabilities corresponding to the possible emotion classes

Train Loop
The train loop is composed by two function called *train video model*, *video predict.*

- *train video model(train_path, train_dir_name, test_path, test_dir_name, emotions_to_idx)*: This function trains the LSTM model on subtrain data and validates it. So, the following steps are implemented:
  1. Loading of features and the corresponding labels from file, if it exists, otherwise they are extracted
  2. Initialization of the model, defining the input dimensions and the batch size
  3. Training and validation of the model, saving the best version of the model on the validation data

- *video_predict(test_path, test_dir_name, features_file_name, emotions_to_idx, cm_title):* This function is used to predict labels relative to subtest and test data using the video model already trained. So, the following steps are implemented:
  1. Loading of features and the corresponding labels from file, if it exists, otherwise they are extracted.
  2. Loading the pre-trained model
  3. Prediction of the probabilities on test data
  4. Plotting the confusion matrix for Test Set

3.3 Audio-Video Model

The Audio-Video Model is used to recognize human emotions through videos and relative audios.
In this model, the ensemble learning technique was implemented using the stacking approach. Specifically, the classification probabilities for the 6 classes (6 emotions) generated by the two models were used as input for this meta-model. This model's task is to recognize patterns in the probabilities provided by the base models and optimally combine them to improve the accuracy of the final predictions.
At first, a list of video-audio file is obtained from the directories of the two datasets. Then, to predict emotions from this audio-video, various steps are followed that are explained below.

Preprocessing of the dataset
The list of paths are processed as follow:
1. Division of the total paths into train data, used to train and validate the audio model and video model, and test data, used to validate the audio-video model.
2. Division of the train data into subtrain data, used only to train separately audio model and video model, and subtest data, used to validate the audio model and video model. In addition, subtest data is used to train the audio-video model, taking as input the predictions made by the individual models mentioned above.
3. Training of video model on subtrain data
4. Training of audio model on subtrain data and obtaining the subtest audio data loader and subtest labels
5. Obtaining video prediction on subtest data and on test data
6. Obtaining audio prediction on test data and on subtest data and obtaining test labels

After obtaining the **dataset** (composed by probabilities predicted by audio model, probabilities predicted by video model and relative labels) and **dataloader** there's the model.

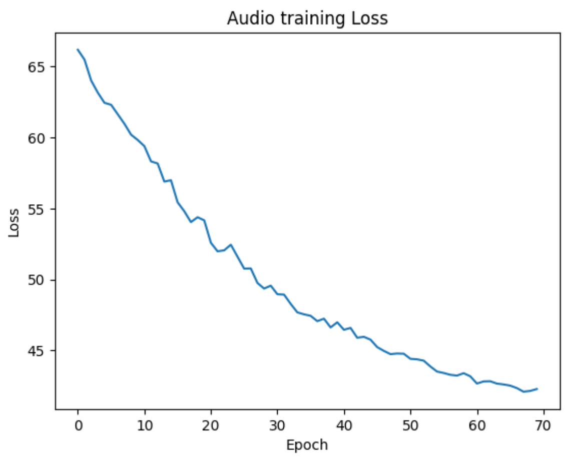
Model
The model is a multilayer perceptron (MLP) developed with Pythorch library designed to classify data based on extracted features. This model is implemented with multiple linear layers, ReLU activation functions, and dropout layers to prevent overfitting, making it suitable for tasks requiring classification based on video and audio features.
Functioning of the Model:

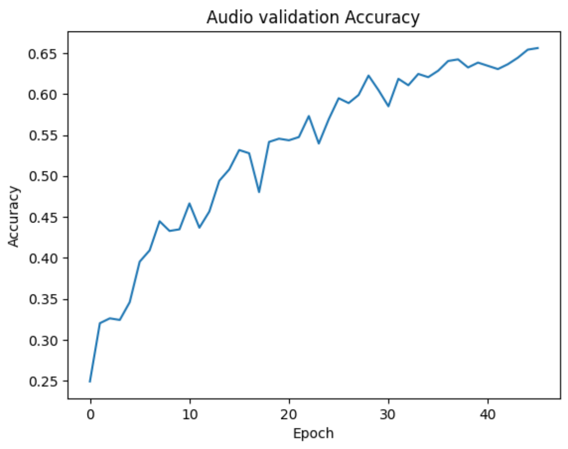
Input: batch of feature vectors
Fully connected layers and dropouts: the vector passes through a series of fully connected layers, with dropouts to prevent overfitting
Output: the output is a vector of values corresponding to the emotions predicted.

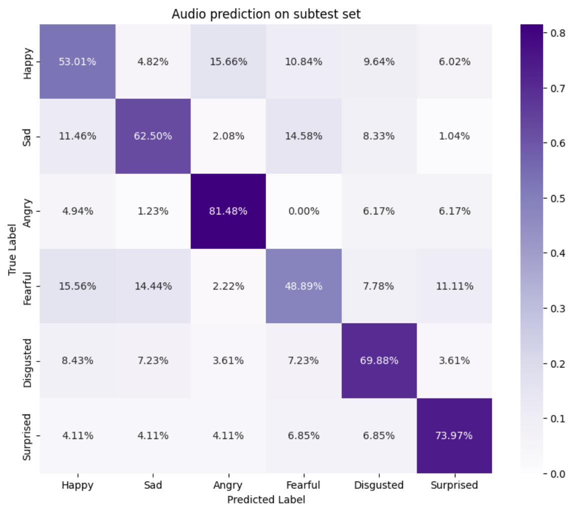
Train Loop
The train loop, defined as a method of the model class, is composed by the following steps:
1. Initialization of the model and confusion matrix
2. Setting the loss and optimizer
3. Training loop, that iterates over the batches of the subtest dataloader, calculates the total loss
4. Validation loop, that calculates the best accuracy to save the best audio-video model
5. Plotting the confusion Loss graph, Accuracy graph and confusion matrix

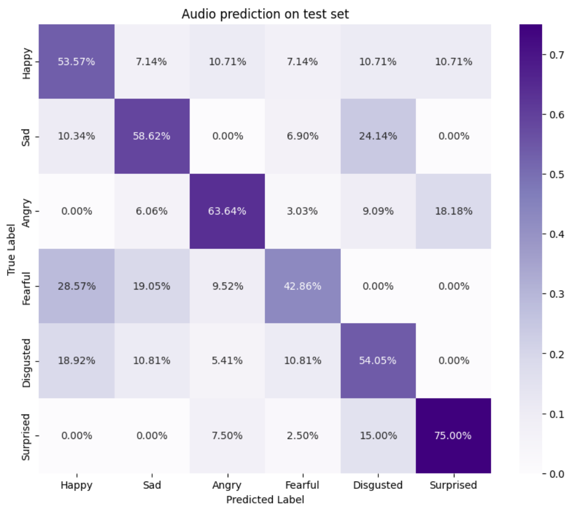
4 Results

4.1 Audio Model

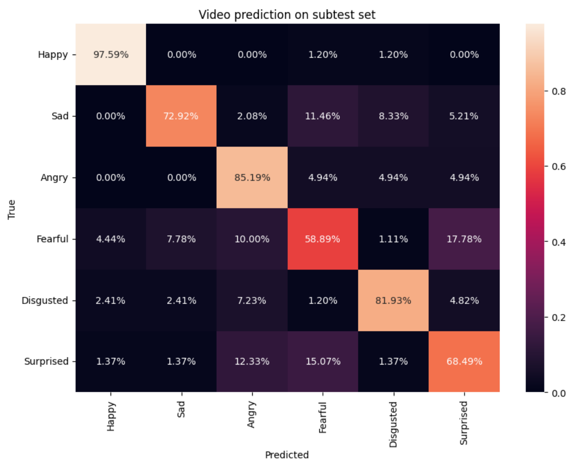


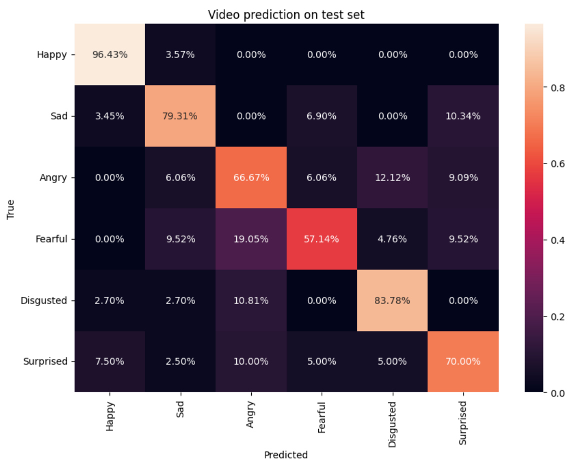




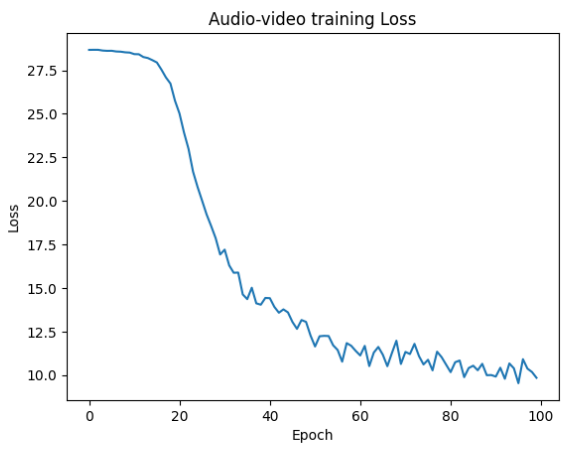


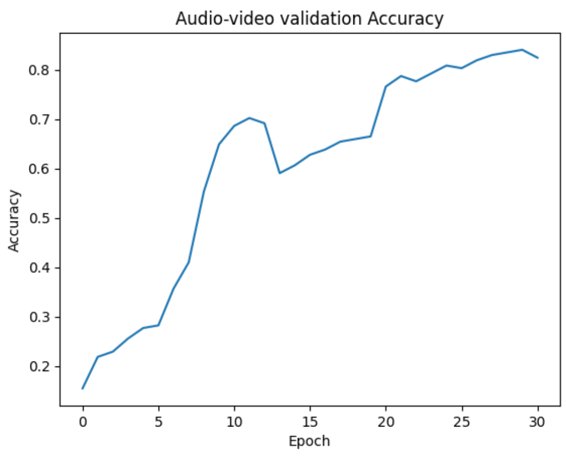
4.2 Video Model



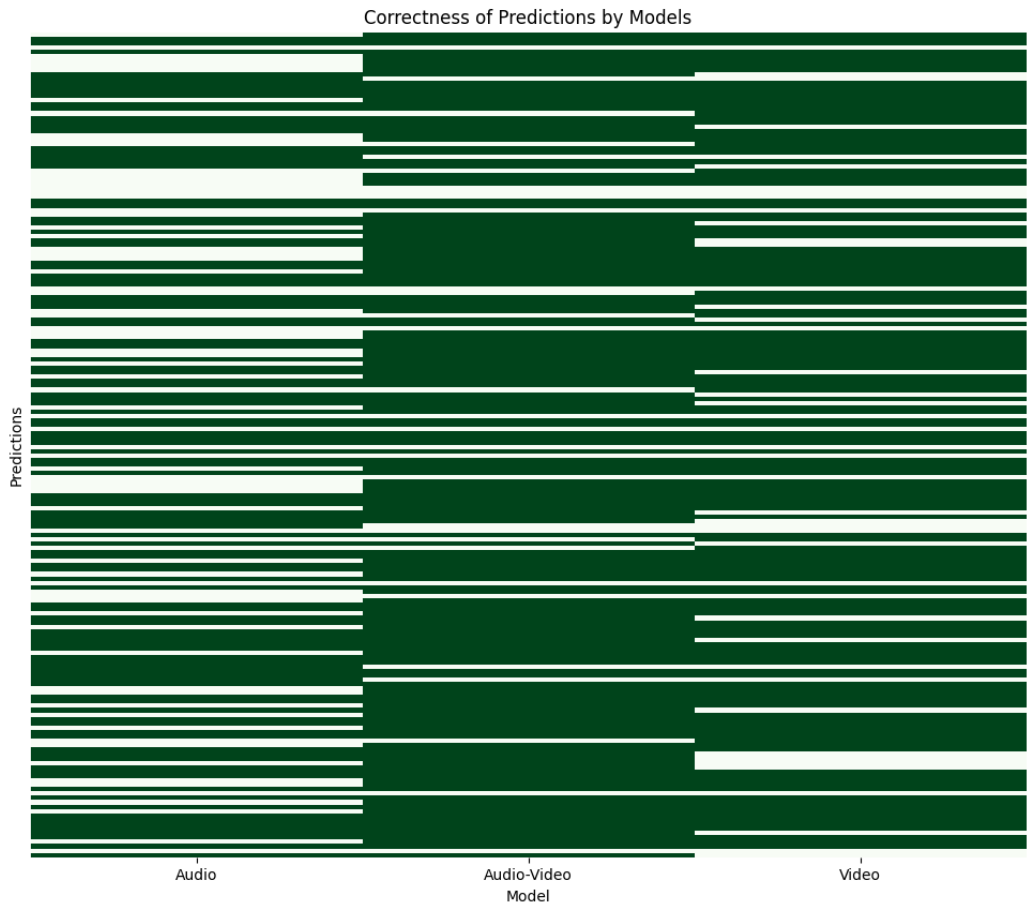


4.3 Audio-Video Model

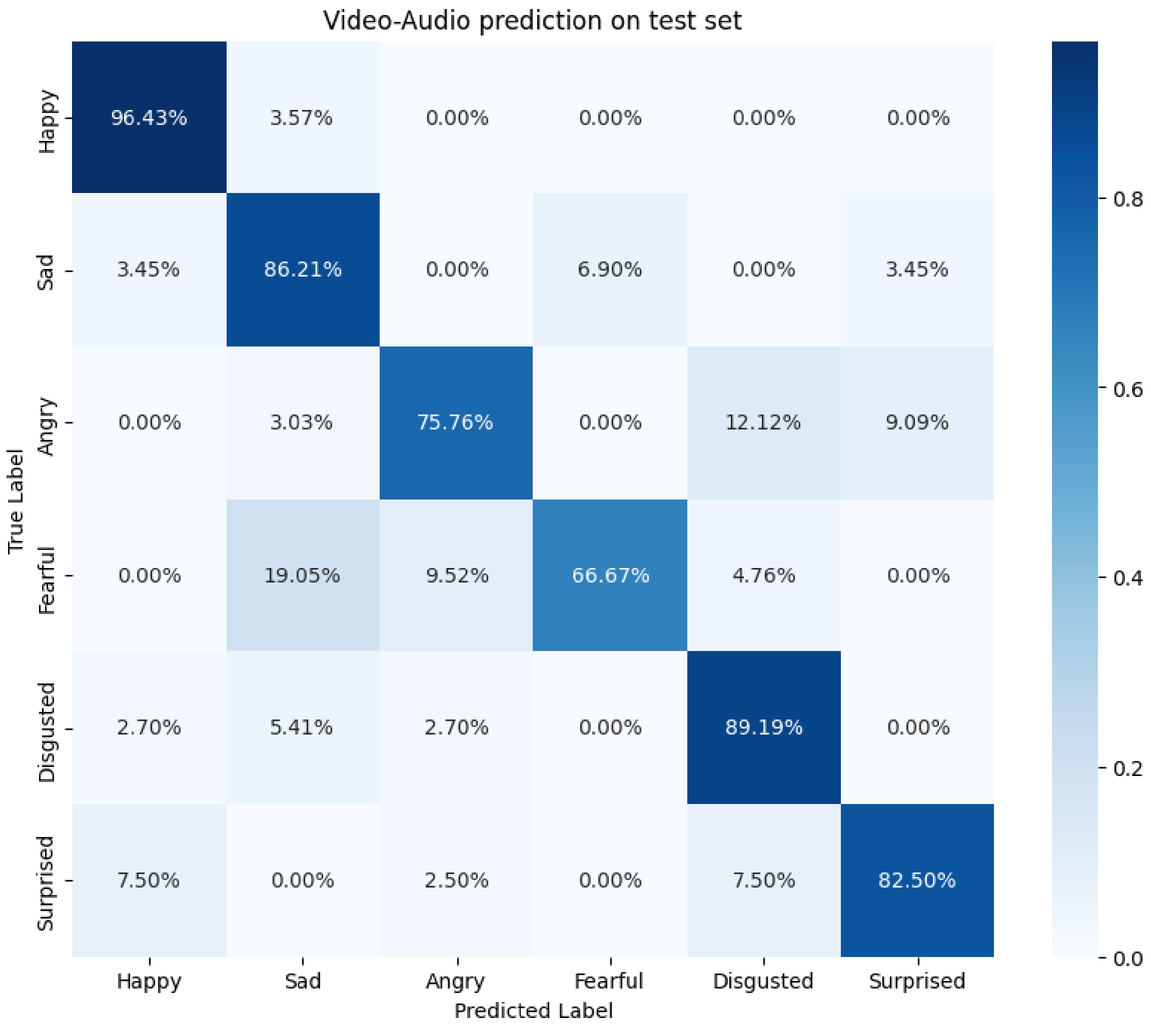




As shown in the graph below, making the prediction on both audio and video, integrating the two individual models into the final model, is very productive as the overall model actually manages to correctly predict more emotions than the individual models.



In conclusion, from the graph below it is evident that the accuracy of the model is 84%. The accuracy of the model would likely improve if more data were available to train and test each models.


Video-Audio prediction on test set
True Label
Predicted Label
Happy
Sad
Angry
Fearful
Disgusted
Surprised