

## Milestone 2 Report

### Current state

We are currently on track for this milestone as we have been able to complete all the specified components listed in the proposal for Milestone 2. The tasks this week included building the front end for the web app, improving the model to get higher accuracy, deploying the final model into the django code in order for the image to be analyzed and return the predicted mood, and creating a song recommendation list. Furthermore, we have been able to complete the facecam capture portion which was a goal for milestone 3.

The frontend was developed using HTML, CSS, and Javascript. Pictures of the current website are uploaded to the github repo section for M2. In addition, we used Bootstrap to support our website design (<https://getbootstrap.com/docs/4.3/getting-started/introduction/>). We embedded a Spotify player to our webpage so that users can listen to the recommended song for 30 seconds directly on our website. To listen to the full song, users need to log in to their Spotify accounts.

The model has been tested using the dataset created in milestone 1. We improved the model by experimenting with different components such as learning rate, momentum, step size, gamma, and minibatch. In addition, we added and removed layers to see what would have better results. We are currently able to achieve an accuracy of 88% after training 10 epochs. The goal in the future is to achieve an accuracy of 90-95% by the end of the project. The model code and testing is written in ModelSkeleton.ipynb.

For the model deployment, we were able to download and use the model within the django code using `torch.load()` to reload the model that was saved from the jupyter notebook. The model takes an image as input and returns its prediction accordingly.

The website is able to capture pictures of the users face with a click of a button and does a post request of a base64 version of the image to the backend where it is converted to an openCV image for the model to use.

### Current challenges

Our current challenges include linking the spotify correctly to the website, ensuring the model is able to return predictions correctly and accurately. As well, the main challenge that will hopefully be fixed for the next milestone is ensuring that the website is able to run properly rather than just locally.

### Team member's tasks

#### 1. Deepan Chakravarthy

I was responsible for testing and improving the model that was developed for the first milestone. Firstly, the sample size was increased through relevant data augmentation. Through researching previous studies on facial sentiment analysis, grayscale was used frequently as it highlighted more features on images of faces. The perspective transform was also used as it is expected for the webcam to capture images that are not perfectly symmetric. The use of the grayscale and perspective transform increased the accuracy by 2% after 10 epochs to 88%. Further, I explored the input space by tweaking the hyperparameters such as learning rate and momentum.

The future task includes experimenting with other layers as using a sequential container with 3 linear layers and 2 non linear layers decreased accuracy by 2-3%. As the model has already reached an acceptable accuracy, I will be working closely with others to integrate the model into django.

## **2. Aryan Gandhi**

For this milestone I worked on some of the front end web development and linked the webpages to the django code in order to receive requests and render the correct html pages accordingly. In addition I worked creating allowing the code to use the webcam and capture the image of the user when the button was pressed. I was able to use a POST request from a javascript form in order to send a base64 version of the image to the django request where I converted the number back to an image using numPy and OpenCV and thus allowing the model to receive it. I was able to test this using the console on google chrome and write the image to a folder to see if the exact same image was sent correctly from the webcam to the program.

The future task for me includes being able to link the backend to the google cloud deployed application and receive the webcam picture from there as well. As well help out with making the frontend better and ensuring we catch all the edge cases that may occur. And maybe start designing a logo for our website.

## **3. Steve He**

I managed to save and integrate the final model into our django webpage using pytorch. So that it takes an image (in this case captured by a webcam) as input and makes predictions about user sentiment. And based on the user's mood we present different webpages and playlist.

My future task includes:

- Deploy the webpage on google cloud so that others could successfully visit our website and receive their own playlist.

## **4. Arya Phan**

I was responsible for building the front end of our web app using HTML, CSS and Javascript. Additionally, I used Bootstrap to support our web designs. I have completed building two web pages so far: the homepage and the output page where we show users our recommended song. These web pages are rendered in the Django framework (views.py) with their appropriate URLs.

My future tasks include:

- Improving user interface. Specifically, displaying users' taken images and allowing them to choose to retake or submit the photo
- Selecting an appropriate playlist based on the prediction of our model and then displaying the playlist in a Spotify player on our web page
- Designing the 'About' page