

Milestone 1 Report

The main goal of milestone one is to collect data and organize them into datasets. We have split the team into two groups separately, with one responsible for the mood classification of music and the other for “emotion” detection of the faces.

Current State for Team "Emotion"

Team “emotion” started off the project by following the instruction proposed in the project proposal (using keras to parse data into training and testing datasets). Since the original reference (related work) involves keras as the API and we are not familiar with that ,we found other similar methods that can handle the same tasks in milestone1 by referring to a new github link [1], instead of the old one mentioned in the proposal. Using “pandas”, we were able to read the csv file from the fer2019 emotion classification dataset. By employing “numpy”, we reshaped the data in the “pixels” column as a 2D array. With the help of “cv2” we then resized it to size=(48, 48). “Numpy.expand_dims” then enabled us to decrease the dims of the 2D array back to 1D, and “pandas.get_dummies” helped us retrieve the label of the dataset (happy, angry, .ex) from the “emotion” column. With the newly parsed datasets containing both label and data, we used “torch” to split the sets into training and testing datasets.

Current State for Team "Music"

For the music portion of our first milestone, we were able to import all necessary packages to create our music dataframes and classified them by mood. We used pandas to create two dataframes, one with appropriate tracks and another including each tracks’ audio features. We then merged them together to create the final data frames which we converted to csv’s. When creating said data frames, we imported Spotipy to help retrieve the tracks and relevant corresponding music data. We modified some of the track information such as removing unnecessary data and additionally labeled calm, energetic, happy, and sad music with their respective moods. The data included in our analysis contains each songs’ danceability, acousticness, energy, instrumentalness, liveness, valence, loudness, speechiness, and tempo. Some excessive data that we removed included the type, key, mode and time signature. Finally, we imported certain libraries to help with the process of splitting the dataset into training and testing sets. We decided it would be efficient if we had individual sets for the mood as well as the features (training 80% testing 20%). When splitting the dataset into its training and testing portions, we used the MinMaxScaler to scale the features within a given range. The MinMaxScaler will initially fit to data and then subsequently transform it. Some challenges we faced during the milestone was learning how to use Spotify because we have never used this library previously. We faced a problem where when we tried to access Spotify music, we were requested an authorization code flow, however,

we were able to find the appropriate Spotify URL. Everything is currently going as proposed, we were successfully able to construct an appropriate dataset with music depending on mood.

Conclusion

So far, both teams ended up parsing the datasets into trained and testing datasets which marked the end of milestone1. There are no feature changes as we are only preparing data to be used in the model, everything goes as planned.

Note: In the proposal, it suggests that we have 200 samples for each music mood, however, due to the duplicates in the datasets, team “music” retrieved 300 samples for each mood, so we could have the data reduced down to approximately 200 samples eventually.

Work Distribution

Yisheng: Imported libraries. retrieved dataset from kaggle, parsed and organized the datasets of team “emotion”, helped find related work

David: Wrote the report of team “emotion” and assembled the two team’s reports into a final milestone 1 report, searched for the dataset of team “emotion” online (kaggle), helped find related work

Jack: Added labels to each collection of music depending on mood, split the data set into training and testing sets.

Jacob: Imported necessary libraries, retrieved music based on mood, retrieved data about each track, cleaned duplicates, merged the information into csv.

Reference

[1] O. Arriaga, “oarriaga/face_classification,” *GitHub*, 27-Aug-2018. [Online]. Available: https://github.com/oarriaga/face_classification/blob/2f152a227c528567924fa1c7587a2e6b8eb43309/src/utils/datasets.py. [Accessed: 05-Apr-2021].