Milestone 4 Report

There are two goals in milestone 4. The first one is to continue working on web development and combining components from both team emotion and music. The second one is trying to improve the validation accuracy of the models.

Face Model Improvement:

We've worked on the feedback offered in M3 for improvement on the face detection model. As feedback has suggested, we tried adding regularizer L2 to the layers to avoid overfitting, but it did not improve the accuracy, and we still ended up with approximately the same accuracy (68%). We have also tried a stronger model like resnet50, However, the result even got worse with around 25% validation accuracy. And the reason for not normalizing and centering the images in the ImageDataGenerator function is because the images are already normalized in the data collection stage with a range of [-1,1], and the images provided in the fer2013 dataset have already been centered. Nevertheless, we tried with those settings on, but there is no improvement.

Since we got the dataset from Kaggle, I have looked at the first place of this challenge, which has an accuracy of 71%[1]. And I have also researched many similar projects that worked with the fer2013 dataset, they all came around 66% accuracy and fairly high accuracy for the training dataset. One of them shares a similar model structure with ours, they also used Xception as their base model and modified it. However, they have more layers than ours and got a validation accuracy of 66%[2] Compared to our 69% accuracy, I think we did fairly well on this part.

Current Status of Web Development:

The web system has been successfully set and has been public. This web page can be accessed through our domain name with port 80. Our system is running on the virtual machine held by Google Cloud Platform, and the IP is resolved to the domain: http://cpen291.yslalan.xyz by the DNS provider, Tencent Cloud. Continuing the process in Milestone 3, we combine the facial classification part and Music recommender system part together and integrate them into the Django web development project. This web project has the structure labeled in the diagram below. The left side circle is the front-end part, and the right side circle is the back-end part.

To implement this function we decided to use HTML, javascript to support our front-end web development. The code has been deeply associated with the back-end framework (Django framework). As the user activates the function of uploading images to the server, the front-end system calls the back-end system and initializes a POST request to send a file to the backend. In this system, the ajax library has been applied to the system to build this bridge connecting the front-end and back-end system. This relationship has been labeled in the diagram above.

The trained model weight has been saved locally to the server to increase the speed of processing the request by the users. The results returned from the ML part are rendered to the HTML webpage through the embedded functions of the Django framework. As can be seen in the diagram below, it shows the code structure of the whole system. The trained models have been saved in the “ml” folder, and the webpage is rendered by the html files which are in the templates folder.

We designed two web pages for this system, one is the home page for uploading the image from users’ local disk, and the other is the web page for showing the detected emotion and recommended music. As users select one image from their PC or mobile devices and click the button, “Process my image”, the web will jump to the second web page and show the result computed by the system. In this web page, the detected emotion, music name, the specific hyperlink to Spotify and the album cover image will be displayed on this page. If users want to get more suggestions from our system, they can click the button “Refresh Page” at the page bottom to get the next recommended music according to the emotion detected from the image uploaded by the user.
Website Preview: http://cpen291.yslalan.xyz

Improving Music Mood Model

For milestone 4, the music team mainly focused on improving aspects of classifying. For instance, previously we needed to train the model every time because it would not save and this was extremely time consuming.

However, after some online research, we found a resource that says the KerasClassifier does not actually have a save attribute [3]. It states that the cross-validation function, KerasClassifier, is meant for model checking instead of model building, so we actually had to rebuild the model to account for this oversight.

We built the model again and trained it without the classifier, however, due to this alteration we encountered another obstacle. When training without the classifier, the prediction output becomes a tensor of possibilities of the four moods. Therefore, to

account for this we had to take an argmax of the tensor to determine which mood the song will most likely be. Additionally, the labels had to be transformed to one-hot code labels in order to match the tensor.

Finally, we trained the model with a batch size of 64 with 300 epochs and achieved an accuracy of 82%.

Work Distribution

Yisheng: Worked on web application

David: Responsible for improvement of the validation accuracy of the face detection model, as well as revision and editing of the report.

Jacob: Changed the model of the music_model to save the model which improved the speed of classifying.

Jack: Responsible for the music portion of the fourth milestone report.

Reference

[1] “Challenges in Representation Learning: Facial Expression Recognition Challenge,” Kaggle. [Online]. Available: https://www.kaggle.com/c/challenges-in-representation-learning-facial-expression-recognition-challenge/leaderboard. [Accessed: 16-May-2021].

[2] 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].

[3] “KerasClassifier has no attribute save.” Stackoverflow. 2019. [Online]. Available: https://stackoverflow.com/questions/55814197/kerasclassifier-object-has-no-attribute-save [Accessed: 13-May-2021].