

Current State of the Project:

*****For our latest code, please check out the milestone 2 branch.**

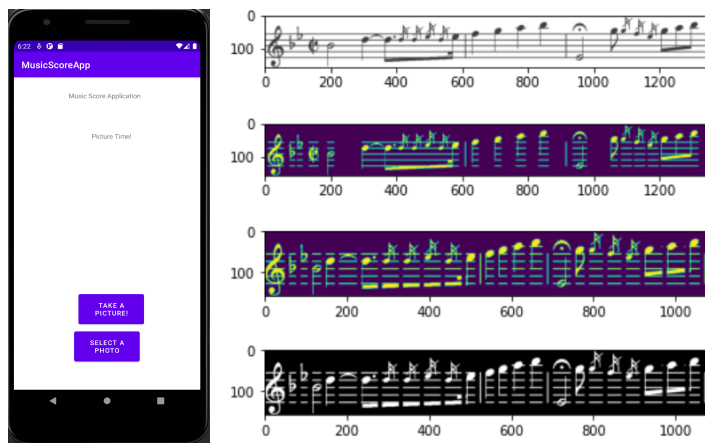
Classification Model & Dataset:

We have designed a custom sequential neural network composed of convolutional, max pooling, and GRU layers—this is our CRNN network and it utilizes the CTC loss function. Currently, our training accuracy has been receiving decent results for a nascent neural network after training it on ten epochs, ranging at around 70-80% accuracy. We hope to develop this neural network even further and train it on more epochs to increase our overall accuracy for future datasets.

For our dataset, the lines of the stave can be detected and most of the notes can be split, including those being placed above the connected notes. As shown below, we display an example of how the scores can be split with white space being filled between for better feature detection on our CNN model.

Mobile App:

An additional feature on our app has now been added where we can now upload a photo from the user's gallery.



Server and Back End:

A linux virtual machine has been set up with the service provided by Google Clouds. The virtual machine is currently hosting an Apache server that can be accessed using the link: <http://35.232.70.229>. The Apache server will host machine learning components and interact with the Android application by receiving web requests from the phone.

What tasks were done:

- **John:** Designed a custom CRNN network with CTC loss function. Trained the classifier model on the current PriMUS dataset and tested it on sample musical scores.
- **Jefferson:** Wrote the code for the implementation of searching an existing image from the device
- **Ying Qi:** Created a tentative test server.
- **Houlin He:** The splitting algorithm is improved (eg. the eight notes, sixth notes are splitted). And the notes being placed above the long notes can be splitted.

Proposal Changes

We only had 1 major change - which is where and how our ML model will run on. Originally we planned to find a place to host a Django server, but Google Cloud Platform proved to be superior as it contained all the elements we needed.

Advantages of Google Cloud Platform include:

- Ability to host a production server as opposed to the development server that is embedded in Django which can only be run as a local host.
- Jupyter Notebook GUI that allows much easier embedding of the ML model.
- Google Cloud Platform had a much better support for machine learning projects overall. For example, it is easy to add or remove GPUs mid project, and there is much more documentation and resources to be found.

Current Milestone 2:

- Configure the CRNN and train it with the dataset. Print the notes on the staff-line with their respective durations and in sequential order with decent accuracy.
- Learned about Django but our research led us to use Google Cloud Platform for our backend
- Added an option on the application to select for existing photos of musical scores in the device

Current Challenges

Classification model:

- For the next milestone, we will work on a greedy decoder. This exists on TensorFlow (https://www.tensorflow.org/api_docs/python/tf/nn/ctc_greedy_decoder), so we will have to replicate its implementation on pytorch. As shown below, the decoder will remove duplicate tokens in a reasonable manner by predicting the best token sequence.



Android App and User Interface:

- The main challenge is to find a way to connect the local data to the server as well as pulling data from the server. This part is likely to be done after the server has been set up.
- Some android devices may not have some photo gallery application that can be called upon to access images stored within a device - in the future we may add an implementation where we directly access the android filesystem when this medium does not exist

Server and Back End:

- The challenge remains on how to properly add the machine learning model to the back and server. This is already made easier by the decision to use Google Cloud Platform.
- Another major challenge is to use service requests to signal the ML model on the server to produce a result.