Introduction

The goal of this project was to benchmark the performance of CNN models' ability to classify geospatial polygons based solely on their geometric shape represented in an image format. Spatial polygon data was converted into 100x100 PNG images and then used to train and score various CNN models. Outputs were compared with published results from literature using a comparable method. As a group we decided to assign me with the project planning, research, data management, and development of generic scripts used to facilitate project implementation. Dan was assigned building custom CNN models, and Limin tackled using pre-trained models.

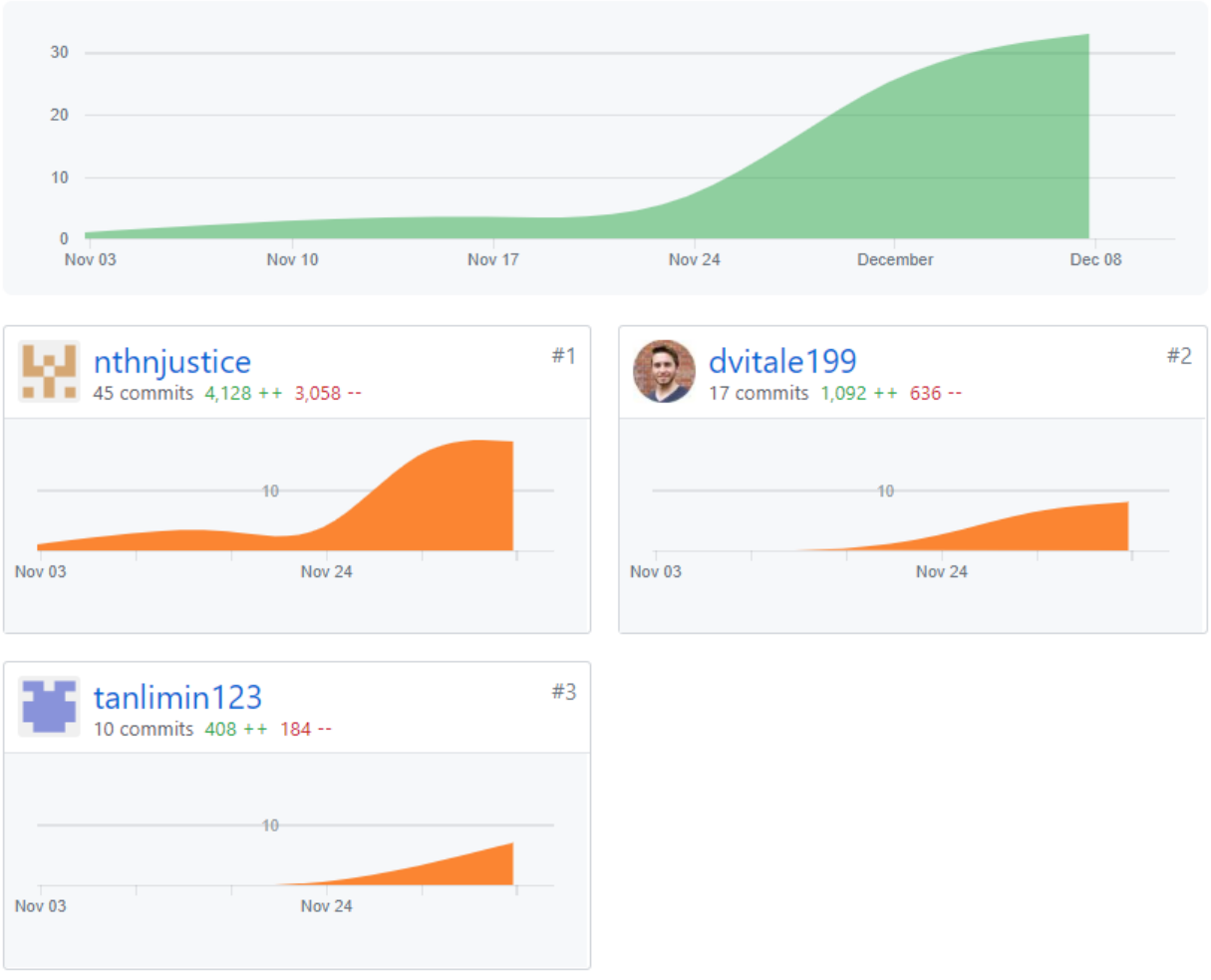
Description of Work

Upholding my designated responsibilities, my early work on the project was extensive literature review, project planning, and coming up with the project narrative with specific, measurable end-goals (achieve comparable accuracies to published results using a variation on existing methods). I spent at least 20+ hours sourcing, cleaning, exploring, and experimenting with different datasets for us to use – everything from species range distributions to land ownership. I upheld my responsibility for writing all data-related code and project helper scripts. The way I designed the project was in a way such that the other group members didn't have to touch any data, data-related scripts or folders in order to work on their models, even as the data sources changed. All they had to do was run a single project initialization script and it would execute all of the data loading, preprocessing, and train/test/validation splits suitable for Keras' ImageDataGenerator automatically. Although I was not necessarily responsible for it, I also

wrote and ran a ton of different models. My implementations led to breakthroughs in prediction accuracy bumps such that all subsequent models written by either of us were based on those architectures. I wrote all of the background and data-related text for the final report. I also put together all of the PowerPoint presentation except for the modeling and results slides.

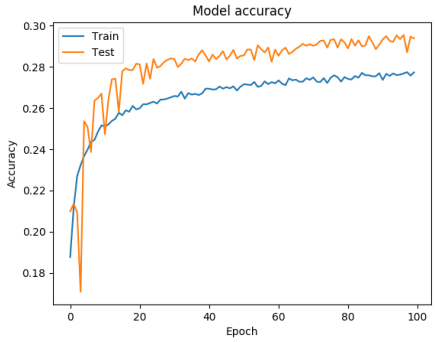
Results

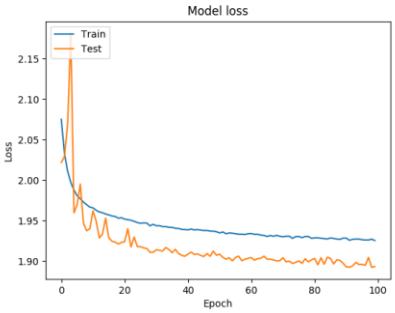
The following is a representation of my code contributions (my primary responsibility for this project):

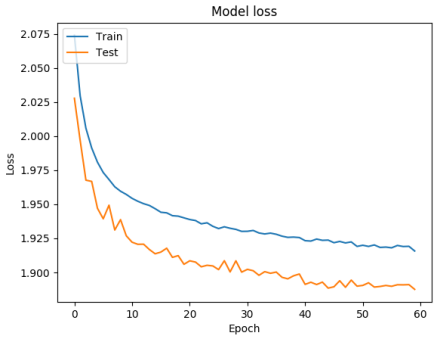


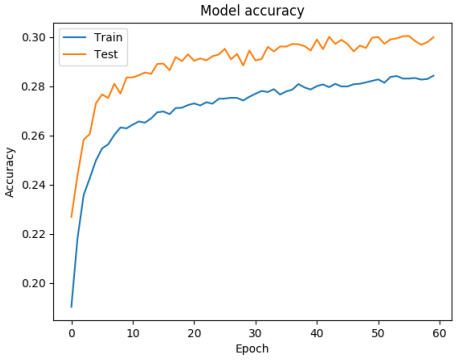
The following results were based on models that I implemented and ran. However, because

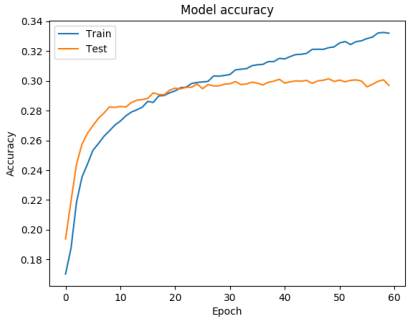
modeling was not my primary contribution for this project, Dan assumed the responsibility of interpreting, understanding, presenting, and documenting the findings of my results. For that reason, discussion and elaboration will have to be found in the other documents found in this project repository (specifically the final report and Dan's individual report).

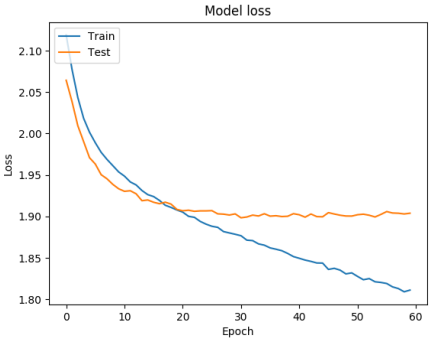


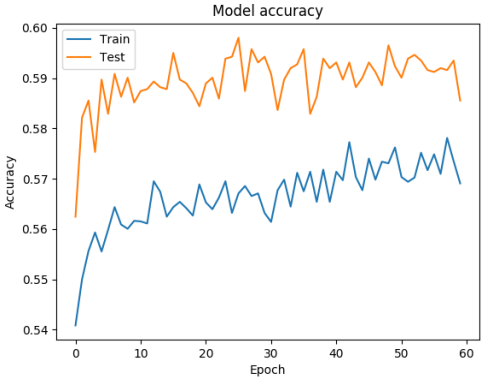


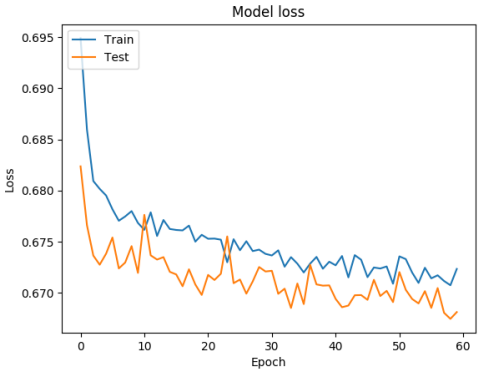


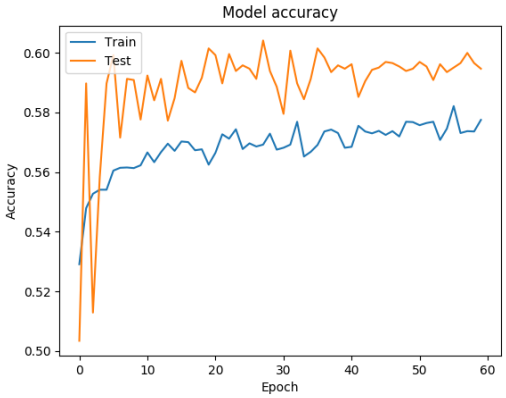


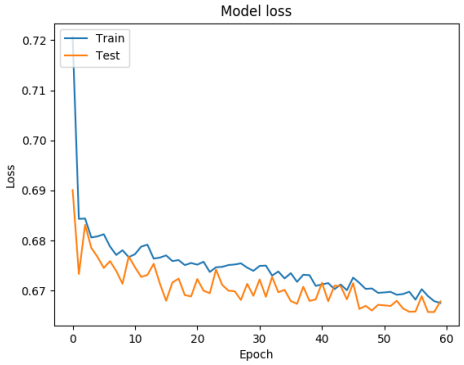


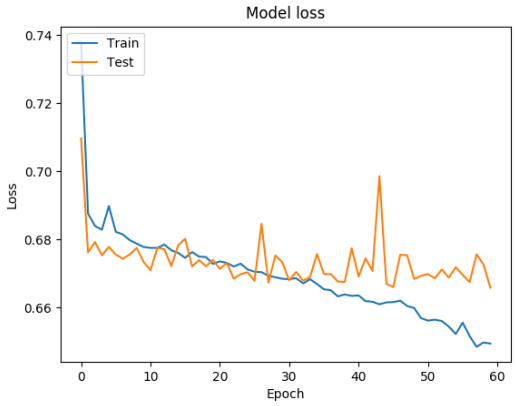


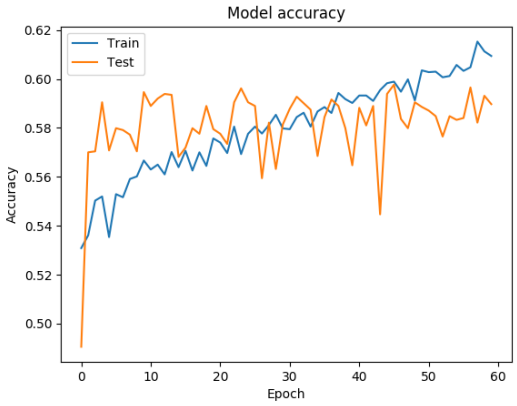

Model accuracy
Train
Test
Accuracy
Epoch

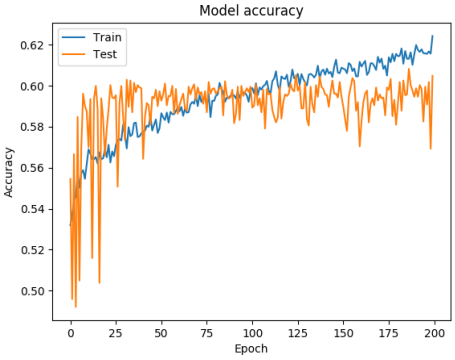

Model loss
Train
Test
Loss
Epoch

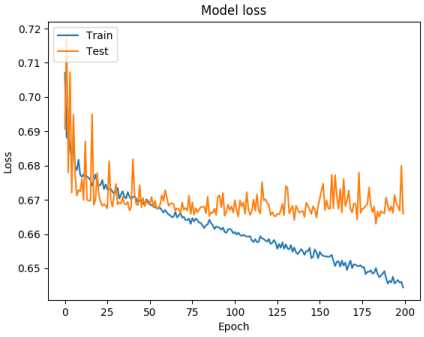

Model accuracy
Train
Test
Accuracy
Epoch

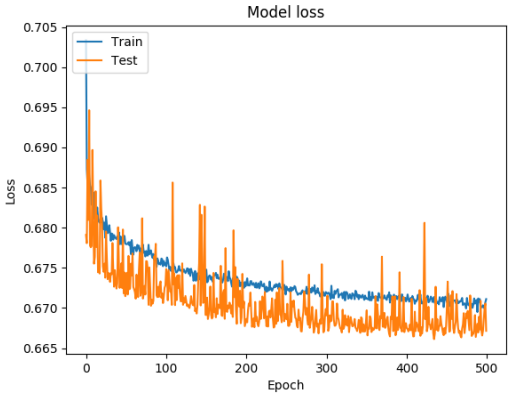

Model loss
Train
Test
Loss
Epoch

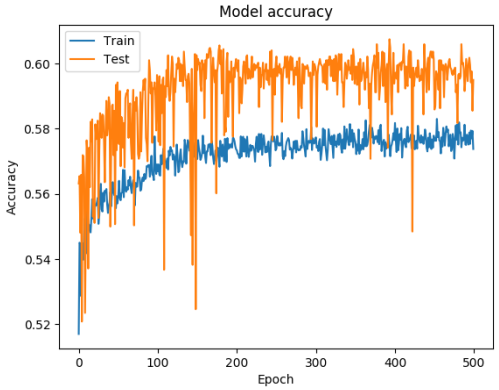

Model loss
Train
Test
Loss
Epoch

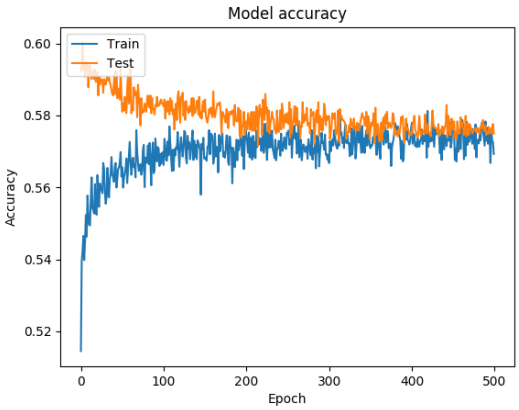

Model accuracy
Train
Test
Accuracy
Epoch

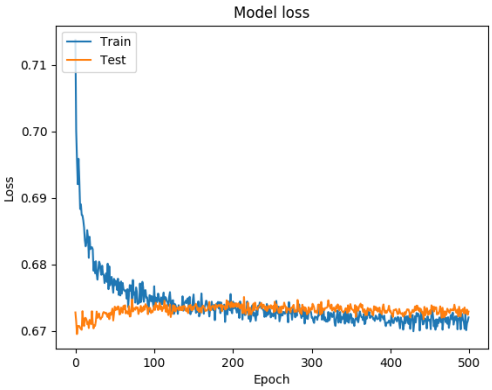

Model accuracy
Train
Test
Accuracy
Epoch

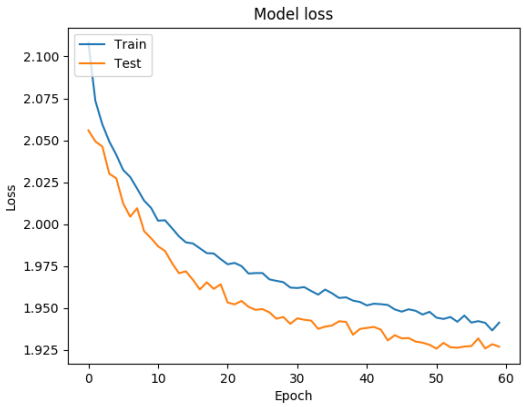

Model loss
Train
Test
Loss
Epoch

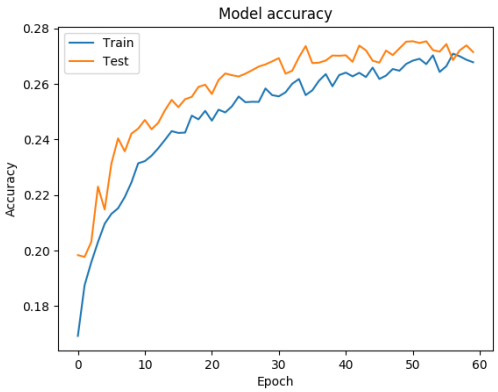












Summary

I have used two machine learning finals as opportunities to explore the possibility of classifying geospatial polygons based solely on their geometry using two vastly different approaches. In my first attempt, using machine learning algorithms trained on feature extractions, I achieved

considerably high prediction accuracies on the data that I used. For this second attempt, using CNN models trained on image representations of geospatial polygons, we did not succeed as much as we had hoped given the data that we used. That being said, I believe there is still opportunity for future research and work on this problem because its application is both bountiful and fruitful. The main conclusion that is abundantly clear to me at this point in trying to solve this problem is that the performance and utility is *extremely* data and context specific. Despite that inherit nature, it could still be usefully and productively applied in the right circumstances, such as internal use at a company or organization that works extensively on a particular type of spatial data.

Copied Code

- Lines of "production" code written (not counting obsolete code/scripts that were deleted): ~521
- Lines of code copied/modified directly from online reference: ~110
- Percentage of code copied/modified directly from online refence: ~21%