**High Altitude Drone Navigation**

NACME Google AMLI 2022 – UARK Team DASA

**PROJECT ROLES**

- Santiago Dorado: Project Manager

- Devin Hill: Programming Lead

- Adrian Whitty: Documentation Lead

- Ayia Ismael: Presentation Lead

**OBJECTIVE**

The objective of this project is to use machine learning models to predict the location of an aerial drone. These models rely on real-time high-altitude to maintain the geolocation of the drone.

**ABSTRACT**

Long-term, we want to use Convolutional Neural Networks in application to Absolution Visual Geolocation to prevent the possible drawbacks and lack of secure navigation when navigating by GNSS. In the long run, taking this model and its subsequent algorithmic modeling and applying it to sections outside of the Washington County area that the current images used for testing and training are based on. In the short term, we want to optimize the models for better accuracy using 12 training sets and 1 testing set for Washington County.

**TASKS**

* **Phase 1**
  + Create a design document on how we will approach creating the model(s)
  + Create a high-level project plan of tasks for each team member with deadlines
* **Phase 2**
  + Obtain and prep the dataset
  + Define and train the model
  + Create the project presentation
  + Discuss acquiring data and exploratory data analysis
  + Justification for the first model
* **Phase 3**
  + Use data to tune the model and validate it
  + Continue the presentation
  + Share iterations of the model tuning with a focus on challenges and questions
  + Get follow-up feedback from both peers and instructors
  + Hold both group and individual meetings with Dr. Rainwater
* **Phase 4**
  + Final iterations on model predictions
  + Review results and prepare conclusions
  + Practice the final presentation
  + Demo. collect, and incorporate final feedback

**DATASET**

The dataset that we will be using for this project is a dataset of 24,000 images from a 30-kilometer squared area within Washington County, Arkansas. These images were low-resolution images provided by GIS Mapping Software. The dataset included images from the same area that range over different years and different ranging from 2006 to 2020.

**MODELS**

For this project, we decided to use Xception and Vision Transformer models. The models that we used were developed and constructed previously by Dr. Rainwater and Winthrop Harvey. Our main objective was to improve the performance of these models.

When we initially ran the model that used vision transformers, the baseline model performance was a 1.0689 root mean squared error, a loss score of 7.9921, and a 49.16% categorical accuracy. While the model’s performance does seem average, the root mean squared error was an indicator that the vision transformer model would not properly categorize and recognize new datasets.

The first thing we did with our vision transformer model, was that we add more layers to the top layer. Below is the code snipped of the added top layers.

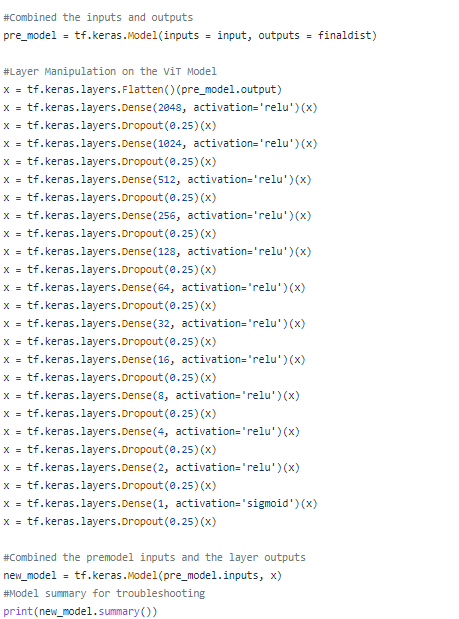


Figure 1. Added multiple layers.

This change in our model drastically decreased the root mean squared error to 0.2661. Another thing that we did was change the number of layers from 6 to 4. This change decreased the number of parameters by 30%. This change decreased the runtime for the program while simultaneously not affecting the model’s performance. The next thing we did was change the patch size from 28 to 16. This small change dropped the root mean squared error down to 0.2647.

With our Xception model, we got a similar base model performance. The Xception model that we worked on included an argument that would pick the layer distribution type, essentially it would pick the model estimator. For our model, we used truncated Cauchy distribution since it repeatedly outperformed the other provided distribution types. We also added ImageNet to the Xception model weights. The final thing that we did was an experiment with our model parameters, which included the initial learning rate, distribution type, and dropout rate.

**RESULTS**

Based on our model’s performance, we have deemed that the model that utilizes Vision Transformers was the best performing model. This was due to the optimal RMSE score and loss score being less than that of the model using Xception. Although Xception did have slightly higher accuracy, vision transformers are optimal for large datasets that include millions of images. Since we only had a dataset of 24,000 images, the accuracy, and performance of the model using vision transformers underperformed due to the lack of data. This however is not a real issue since our dataset was only collected from an area of thirty kilometers squared.

**APPLICATIONS**

The real-world applications of these models can be implemented into various operations. This model can be implemented onto unmanned aerial vehicles which can rely on onboard cameras to retrieve the geolocation of the drone instead of relying heavily on satellite transmitted information that is susceptible to spoofing. Another application can be for transportation. This model could be implemented onto aerial vehicles, and this model could provide the positioning of the aircraft so that the aircraft could be flown autonomously and have another method of verifying the aircraft’s location.

**FUTURE IMPROVEMENTS**

One of the constraints that we had was the number of epochs that we could run on our local machines. Initially, we intended to use the Arkansas High-Performance Computing Center (AHPCC) to run our programs, but due to some time conflicts, we were unable to. If we were able to run our program on the AHPCC, this would have given us the option to increase the number of parameters, increase the number of epochs, and also be able to utilize a high-resolution image dataset.

Another improvement that we were able to partially fix, but still requires some work, is being able to have a requirements.txt that is fully compatible with macOS and Windows. Most of the members on our team were using macOS machines, which we later discovered were not compatible with some of the required packages. The memory issues that we encountered while using macOS were mostly sorted out, but this did take a large portion of our time to complete this project.

**CONCLUSION**

For this project, we were able to create a machine learning model that would recognize the location of the image that was input. Our model accuracy was average, but we were able to overall drastically improve the root mean squared error and loss of both of our models. We decreased the root mean squared error by 75.3% and our model loss by 90%.

We concluded that our model that utilizes vision transformers would outperform our Xception model. This conclusion was reached since the vision transformer had a lower RMSE and loss score which indicates that it would better handle new datasets. Overall accuracy is average, but this is due to vision transformer models benefitting in overall performance by having a large image dataset. Since our dataset included 24,000 images from an area of 30 kilometers squared, our model would perform better if it would be further improved and trained on larger datasets.