FarmNet: Identifying farms from satellite imagery.

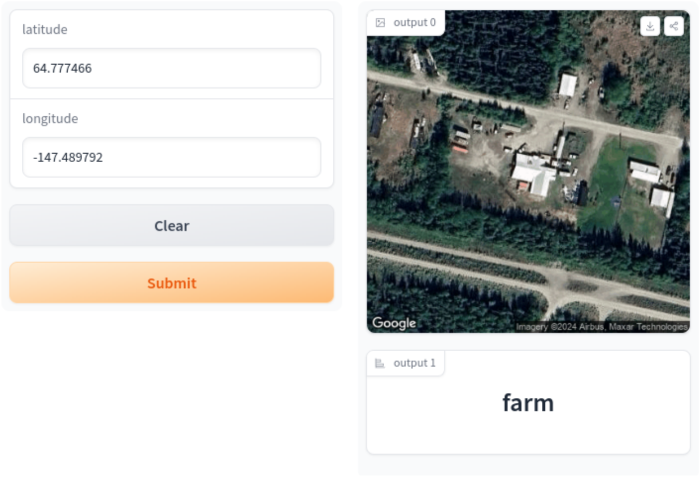
Filip Zawadka Group 51

Abstract

Due to the rise of automation and wealth inequality, I see a rise in factory farms, which contribute a significant public health, especially in neighboring communities. This project attempts to monitor the factory farms using satellite imagery.

1 Website

Making use of a simple CNN and Google Maps Api I created a website to identify factory farms. after imputing longitude and latitude of wanted map point, image of neighboring area will be displayed and identified whether it's a factory farm or not.



latitude

64.777466

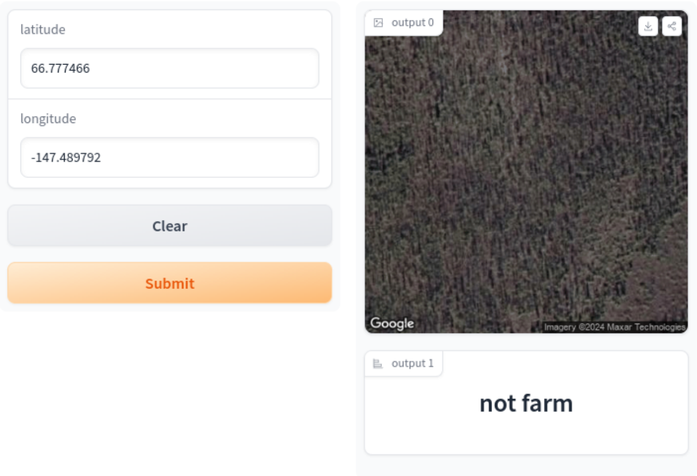
longitude

-147.489792

output 1

farm

Figure 1: Recognized farm.



latitude

66.777466

longitude

-147.489792

output 1

not farm

Figure 2: Forrest recognized as not a farm

2 Data

To generate data I use selenium to scrap data points from counterglow, it's an amazing project where factory farms in the United States are submitted and later verified. I need to iterate over those points to collect latitudes and longitudes of marked points and corresponding labels, which include: "Slaughterhouse", "Broiler (Meat) Chickens","Layer Hens (Eggs)","Cows (Dairy, Meat)","Cows (Meat)","Pigs (Meat)", which provides us with more information about the farm. Points are constantly updated, so to improve the dataset, they should be collected periodically.

After collecting the points I use Google Maps Static Api, using satellite imagery to access the surrounding area of a data point at zoom "17", which should provide the most information about the area, while including enough of detail.

After this process I should have the "true" labels, to generate "false" labels I generate random points and add them to the dataset. I use 3:1 ratio of random points to farms, so it's not overwhelmingly beneficial to choose one label over the other.

3 Model Architecture

I started with resnet18, and fine-tuned it on our data, but very quickly observed that the model over-fitted to the data, seeing validation accuracy getting much lower that the training accuracy. After I expand the dataset, I might come back to this architecture, but for now I decided to use a simpler architecture.

3 Layers of 2d convolutional layeres followed by max pooling and then by 2

linear layers, were used, so we have a simple enough architecture, for the amount of data. I also limited the classification to a binary one, to simplify the task for the model. This led to 0.95 accuracy on the test set.

4 How to run

To run the whole pipeline, first, use the farms_scraper.ipynb, it will collect points from counterglow and append them to farms_scraped.csv file, which all current datapoints

Next, generate_satelitte_images_google_api.iynb, this will make use of the scraped datapints to generate google maps satellite images, it will also add random datapoint images.

Finally, to train the model use train_cnn.ipynb, which will train the model, and add it to wandb model registry. The production tag is set to "latest", so the model should be used on the website, as soon as it reboots.