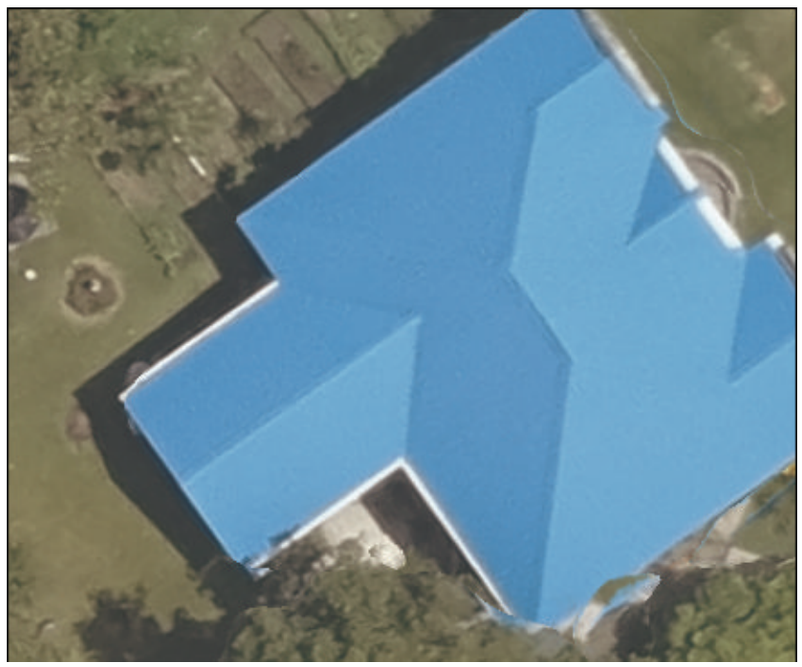
# Predicting Rooftop Quality and Integrity using Drone Image Data

By: General Assembly Data Scientists
John Milne, Sarah Scolnik, Kyle Sokolis, Brandon Walraven



# Welcome

Thank you for taking the time to join us.

# Problem Statement

- **Client problem statement:** The Island of Saint Lucia needs to prepare its homes for the next hurricane season. This tool will use remotely sensed drone images (RGB and elevation) along with GIS layers from OpenStreetMap and CHARIM to assess which homes are at highest risk and which homes might need strengthening interventions (i.e new rooftops, roof straps). The tool will provide the user the ability to click on a building (in the pilot city of Dennery) and see some basic characteristics of the house that could be helpful for the government preparing for the next storm.

- **Project problem statement:** Our project goal was to use the drone RGB image and GIS building data to train a neural network model to identify buildings that may need repairs to prepare for a hurricane. This model could be applied to images of roofs from other locations. The output is linked to the building ID number in the GIS data and can be used as part of a tool which allows the user to click on a building and see the building information.

# Dennery, St. Lucia at a glance


St. Lucia:

- Tropical climate, mountainous topography
- Risks: hurricanes, floods, landslides

Dennery:

- Town & surrounding district
- District population: approx. 12,599 (source: Central Statistics Office, 2011)
- Building inventory conducted in 2015 found most common roof types: galvanized sheet iron, painted sheet steel, concrete (source: “Assessment of physical vulnerability to flood in Saint Lucia”)

Sources:

- http://www.charim.net/stlucia/information
- https://en.wikipedia.org/wiki/Saint_Lucia
- Assessment of physical vulnerability to flood in Saint Lucia. Master’s Thesis by Anne Chinyere Uwakwe, April 2015 (http://www.charim.net/sites/default/files/handbook/otherpages/MSC/Anne%20Chinyere%20Uwakwe-s6012345.pdf)

# Python Library Requirements

## General Data

- import numpy
- import pandas

## Modeling

- import keras
- import sklearn

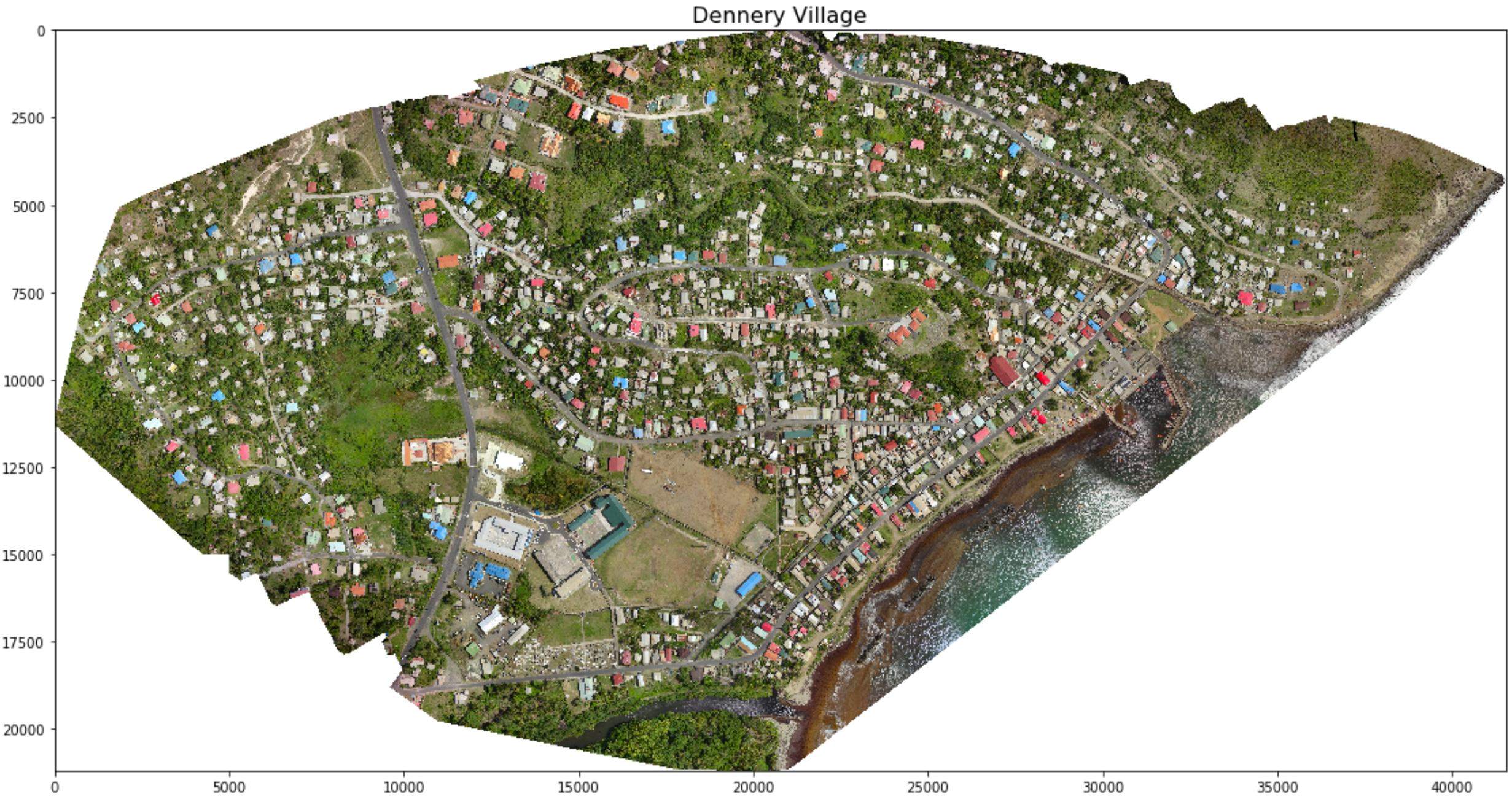
## Geographic

- import json
- import rasterio
- import geopandas
- import pysal
- import shapely
- import pyproj
- import fiona

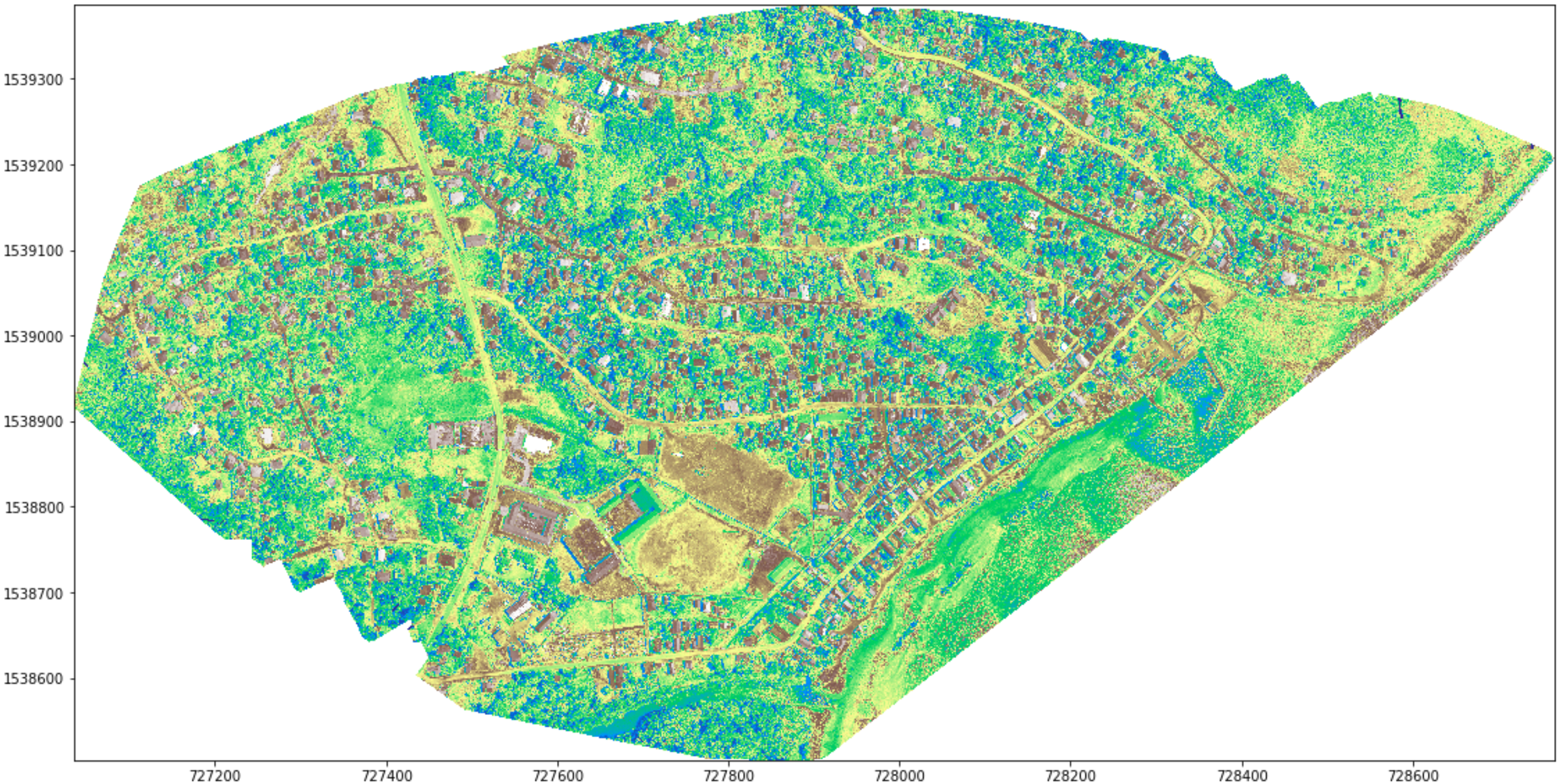
# Initial Approach & Methodology


- **Given ArcGIS files**
- **Found buildings.shp file on CHARIM website**
- **Contains bounding-box of the buildings in the mosaic.tif file**
- **Image processing in Python**
- **Cropped the .tif file at the bounding-box to get our data**

# Image Prep

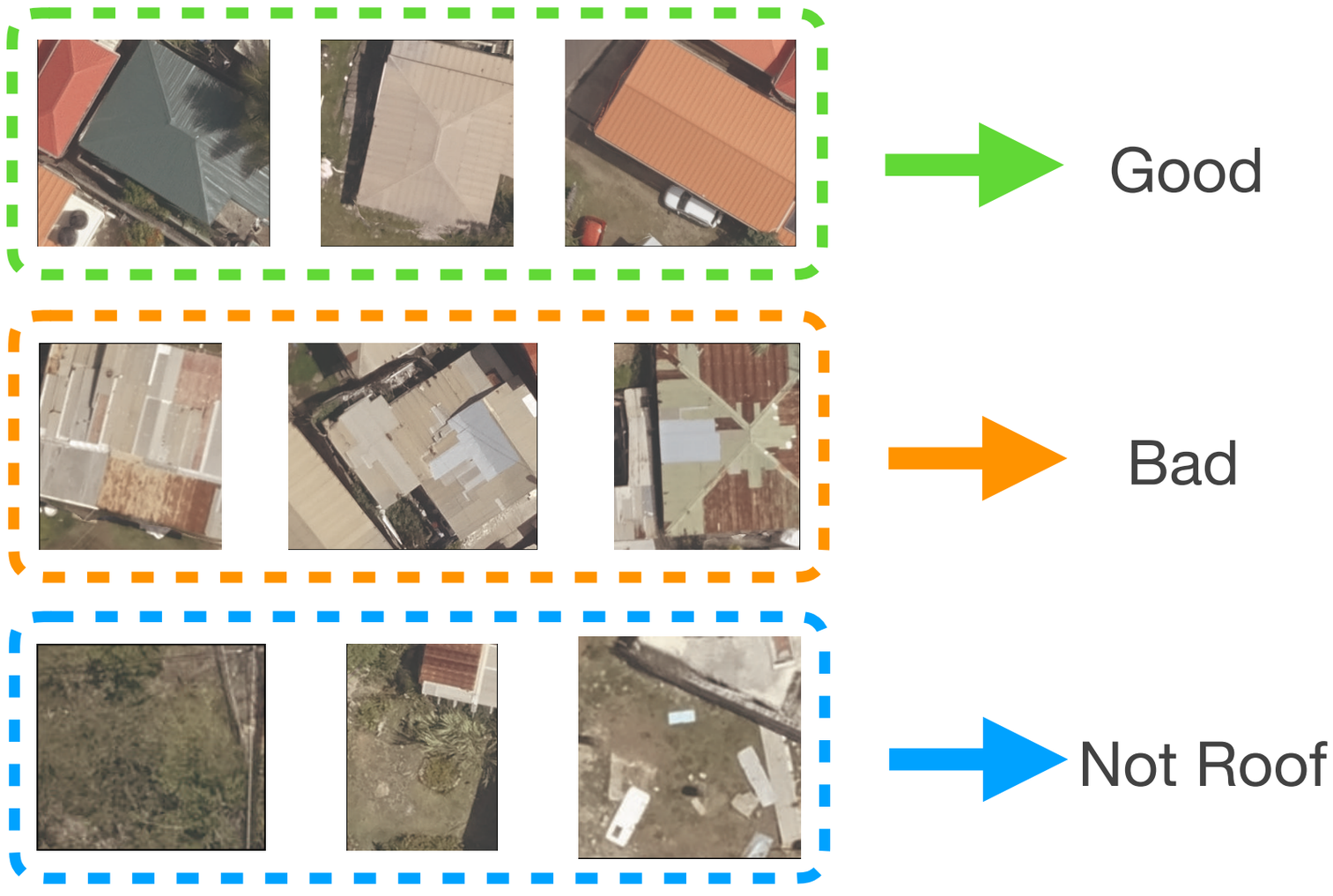

Dennery Village
0
2500
5000
7500
10000
12500
15000
17500
20000
0
5000
10000
15000
20000
25000
30000
35000
40000

# After rasterio processing



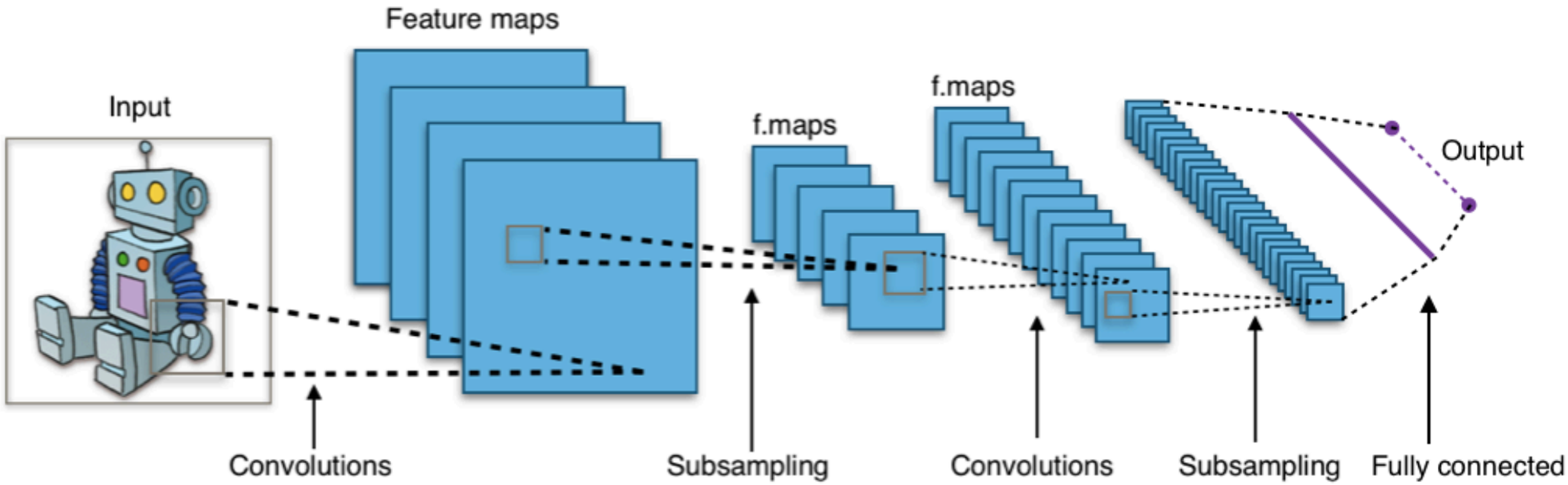
# EDA (Exploratory Data Analysis) and Classification of Images

Supervised learning: We assigned the images to classes so that the model could learn from the training data and then classify test data it hasn't seen before.



# Model Chosen and Why

Convolutional neural networks (CNNs) are the current state of the art for image classification models.




*Source: https://en.wikipedia.org/wiki/Convolutional_neural_network*

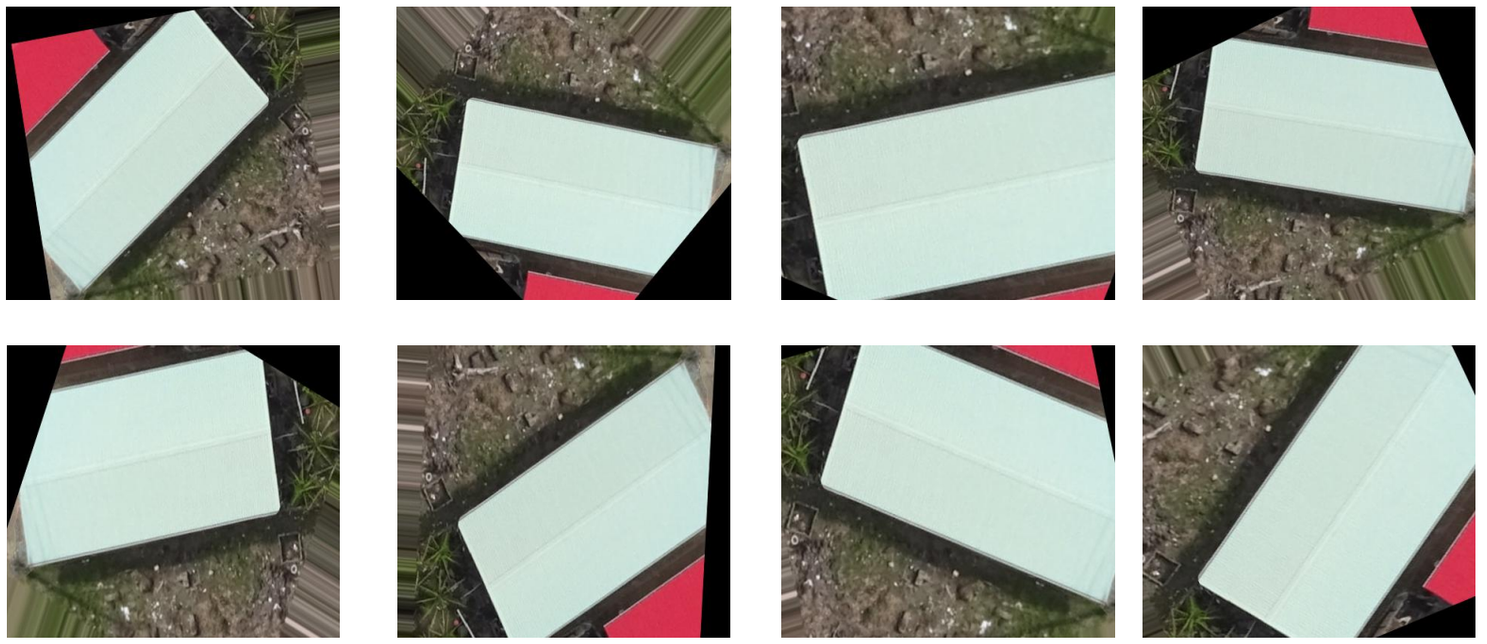
We used Keras, a more user-friendly interface for TensorFlow deep learning

# Problems encountered with size of data set

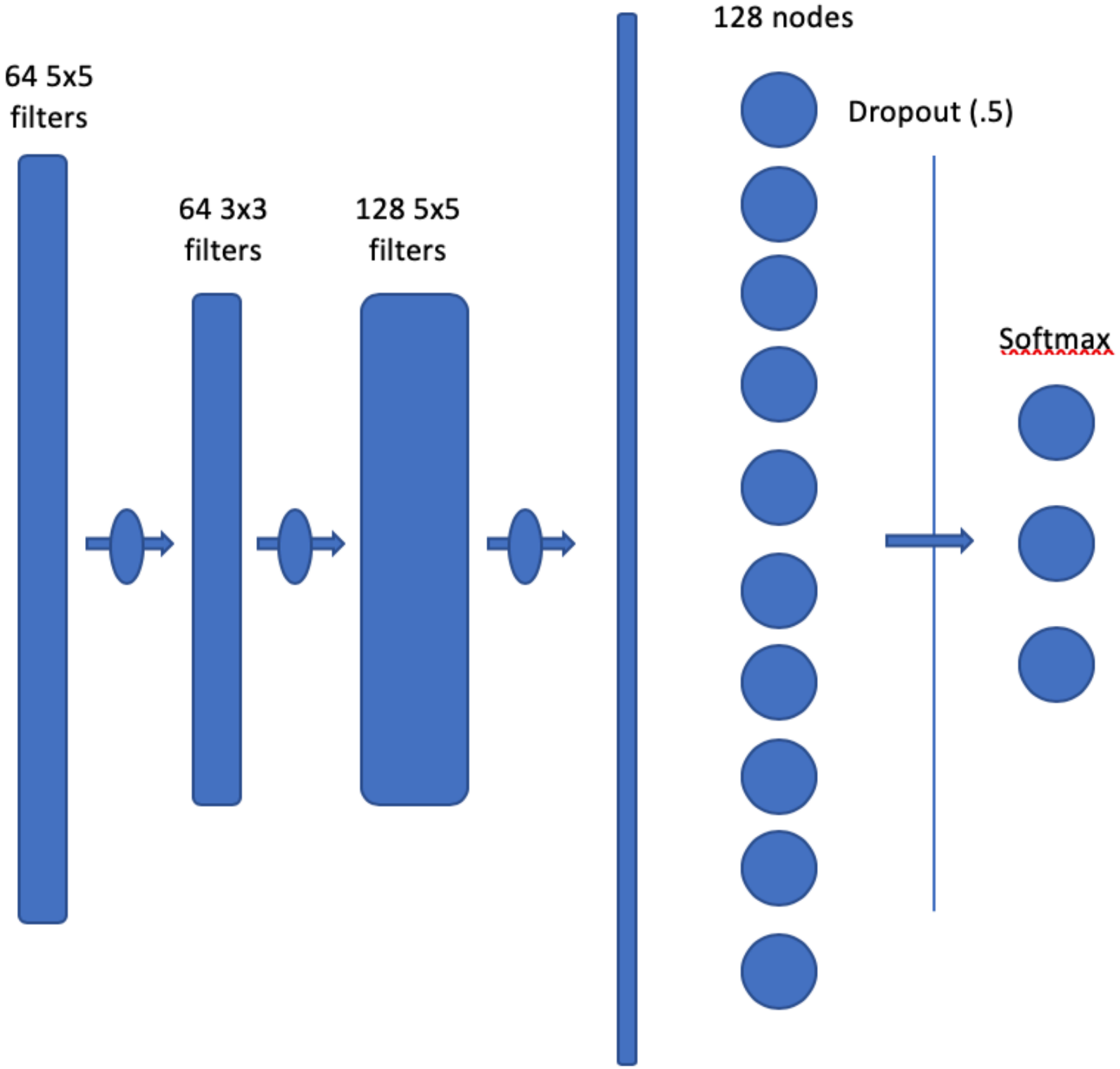
- More training data → better model
- Less training data → overfit model?
- For image classification, 1000 images per class is small
- Data augmentation strategy: random transformations to images

Original

Transformations

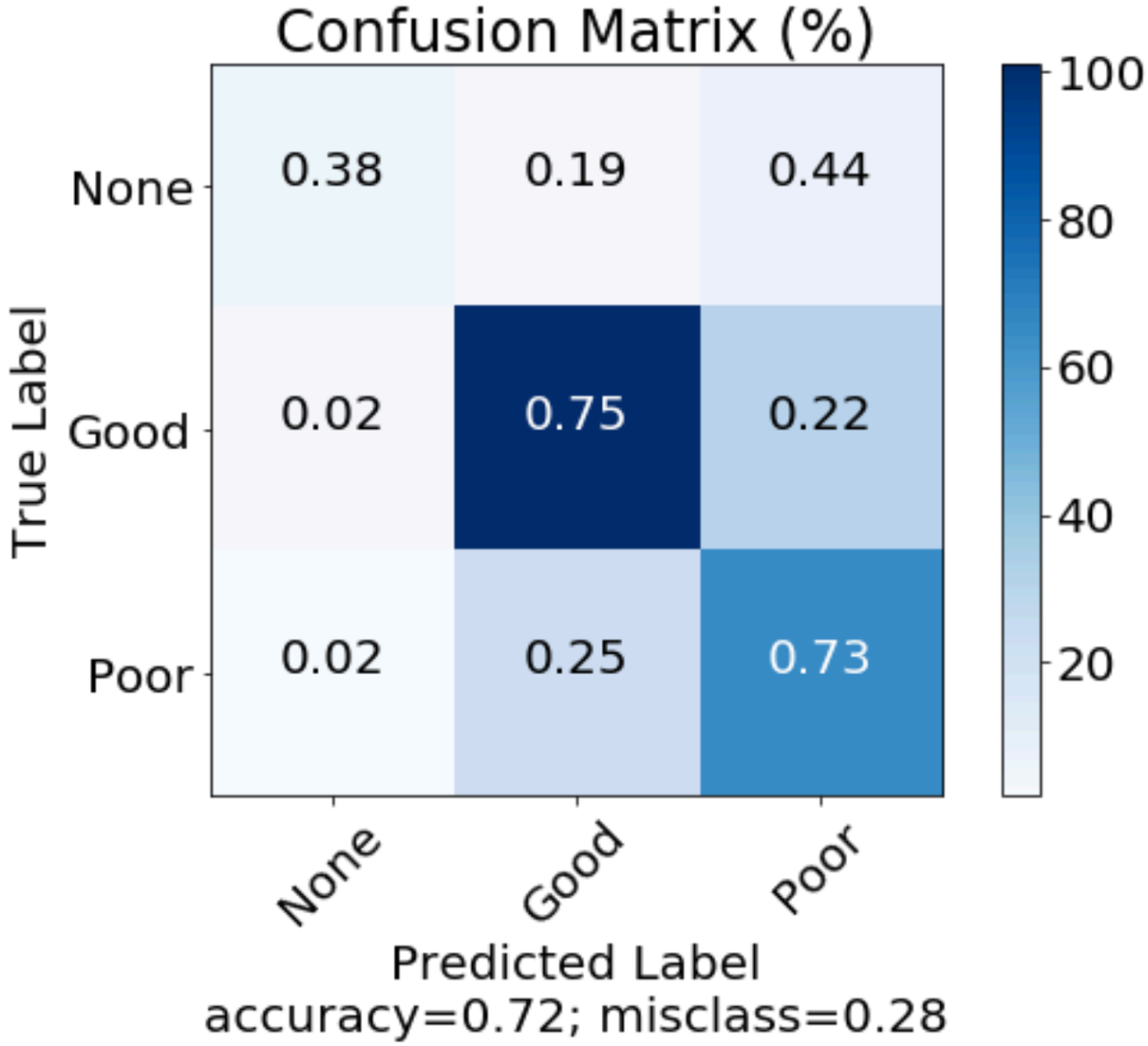


# Model Structure

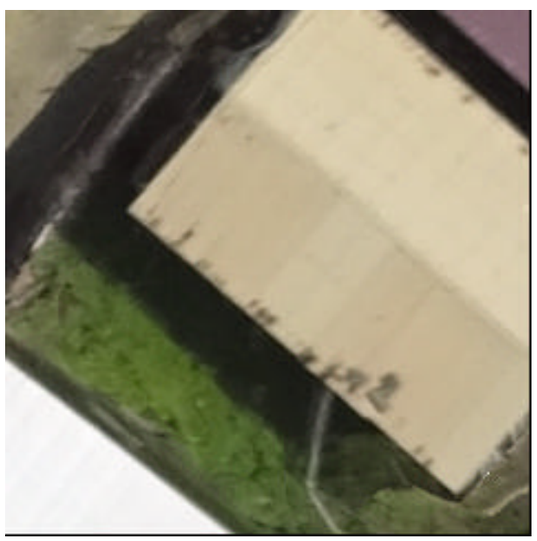


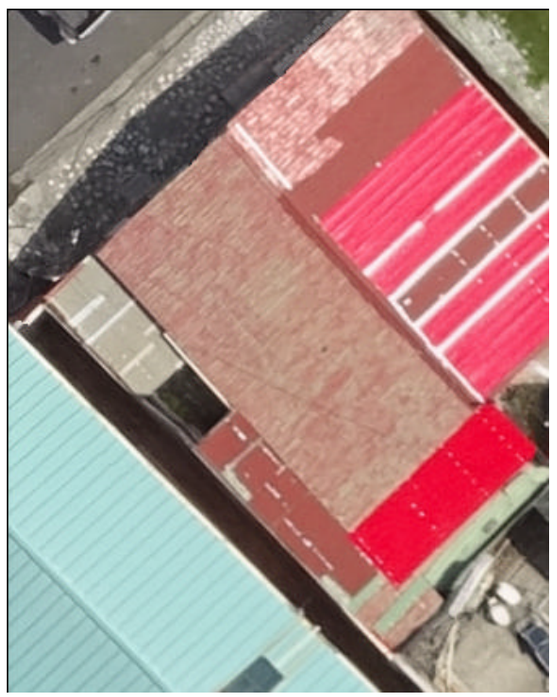
# Model Evaluation

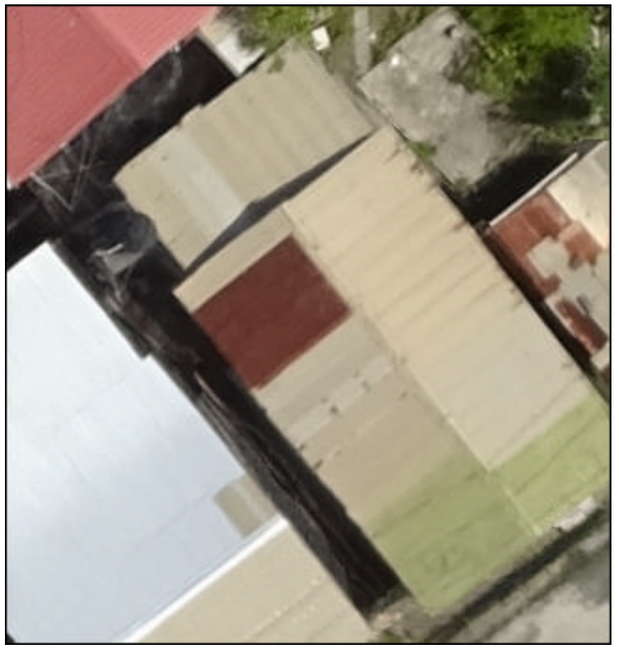
- Model Baseline: 56%
- Model Accuracy: 72%
- Accuracy is probably not the best evaluation metric.
- Recall focuses on the percent of a specific category that the model correctly identifies
- Non-Roof Recall: 38%
- Poor-Roof Recall: 73%

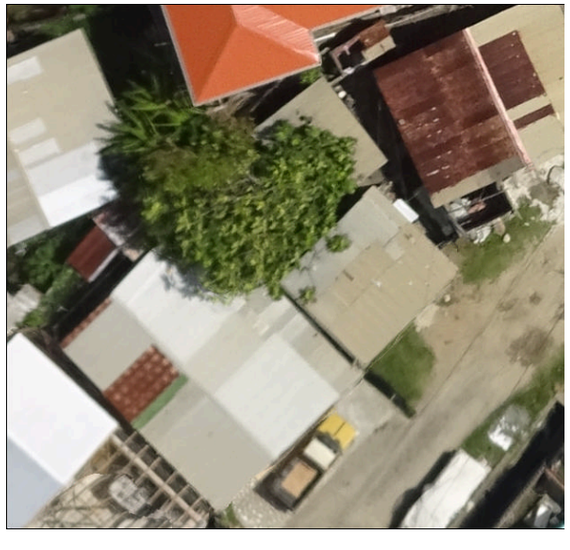


# False Negatives

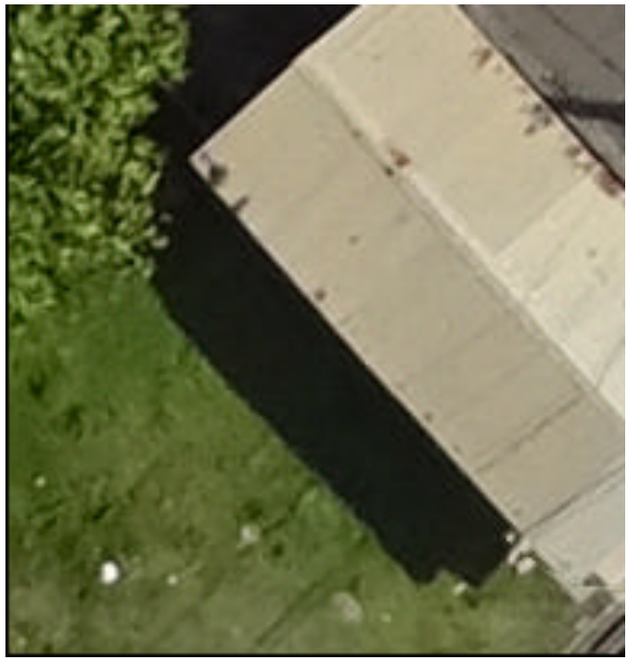


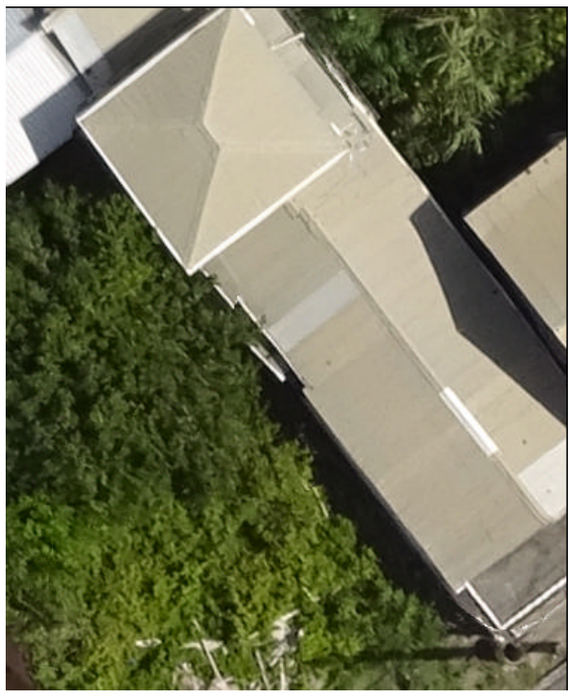




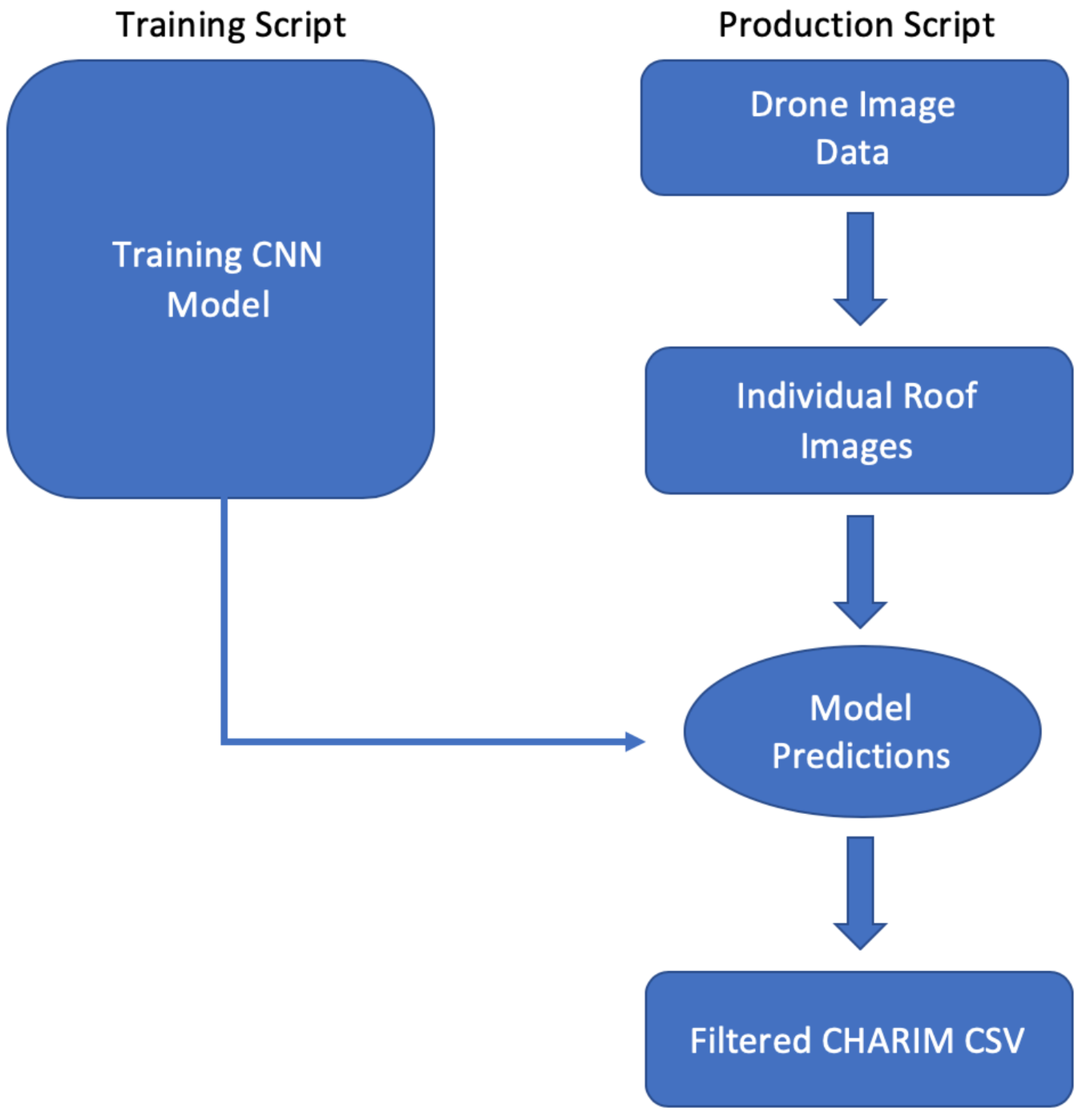


# False Positives





# Model Workflow



# - Road Blocks - Fixes
# - Back-up Plan
# - Further Preprocessing

**Road Blocks:**

- **ArcGIS: Steep learning curve, PC-only,**
- **Data: XY in .tif file | Lat/Long in shape file**

**Our fix:**

- **Interact with the mosaic.tif file in Python**

**Back-up plan:**

- **Crop images manually from ArcGIS**

**Further preprocessing:**

- **Texture**
- **Uniform**
- **Dilation**