



Open Cities AI Challenge

Segmenting Buildings for Disaster Resilience



COMPETITION OVERVIEW

Open Cities AI Challenge: Segmenting Buildings for Disaster Resilience is a competition hosted by DRIVENDATA which aims to accelerate the development of more accurate, relevant and usable open-source AI models to support mapping for disaster risk management in African cities.

The objective of this challenge is to map building footprints only by having access to drone imagery and thus simulating an event of a natural hazard. **Participants are tasked to build an AI model to classify the presence or absence of buildings on a pixel by pixel basis.**

COURSE OVERVIEW

This competition was selected as our **CS513 autonomous agents** class project. As image segmentation was revolutionized by the use of deep convolutional neural networks and is one of the most complex tasks in computer vision, this project was also a decent fit for this class.

INTRODUCTION

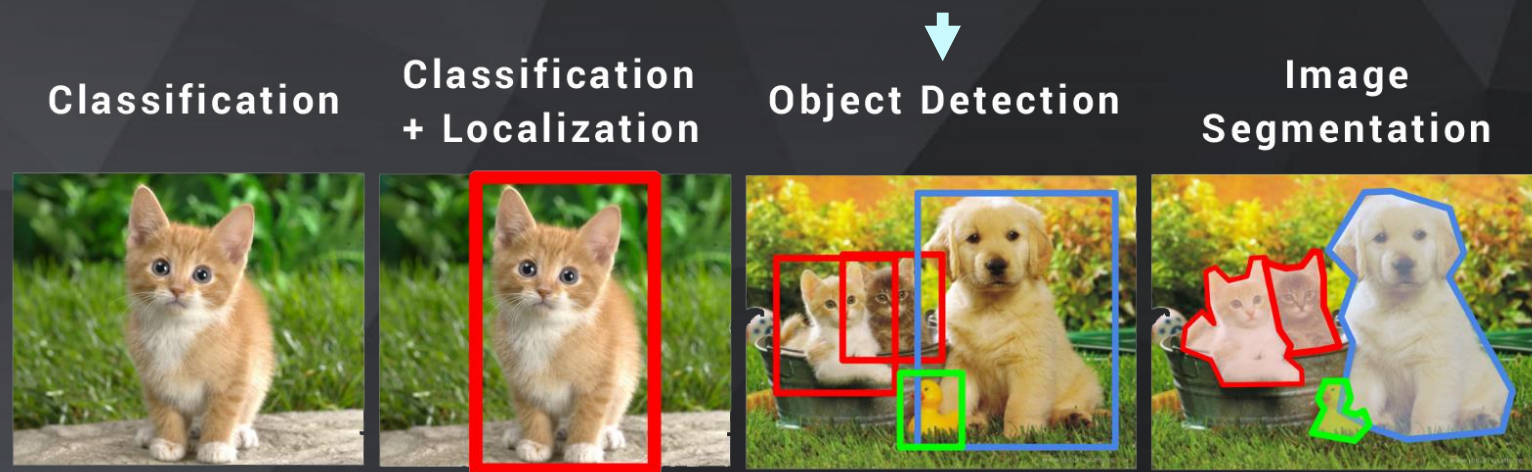
- ➔ **Deep learning:** a machine learning technique that teaches computers to do what comes naturally to humans: learn by example. This is mainly done using **neural networks**, a set of algorithms that are designed to recognize patterns.

Image Classification: output a discrete label of the main object in the image

Classification with Localization: along the discrete label, we localize the object in the image

Object Detection: classify and localize all the objects in the image

- ➔ **Semantic Segmentation:** label each pixel of an image with a corresponding class of what is being represented



PROJECT STATS

Programming Language: Python

Code Format: Jupyter Notebook

Code Environment: Google Colab

Implemented Framework: Keras API

Deep Learning Model: U-net

Metrics: Average Accuracy, Jaccard

Loss Function: Binary Crossentropy

Optimizer: Adam

WORKFLOW OVERVIEW

Our deep learning pipeline can be defined in these 5 stages:

- 1. Gathering data:** provided datasets by competition (`train_tier_1`, `train_tier_2`, `test`)
- 2. Data pre-processing:** we convert our data to a format compatible to the input of our model
- 3. Researching the most appropriate model:** U-net selected (specific type of FCN)
- 4. Training and testing the model on data:** `train_tier_1`, `train_tier_2` datasets for training and `test` dataset for testing
- 5. Data post-processing:** manipulate the predicted results in order to match submission format

A. PRE-PROCESSING

This is the process of converting the “raw” data into a useful format. All the initial data is stored in a STAC. For each scene, the corresponding geojson label file is read and visualized. We then proceed to split it into many lower-resolution pairs of images and masks.

SCENE PREVIEW

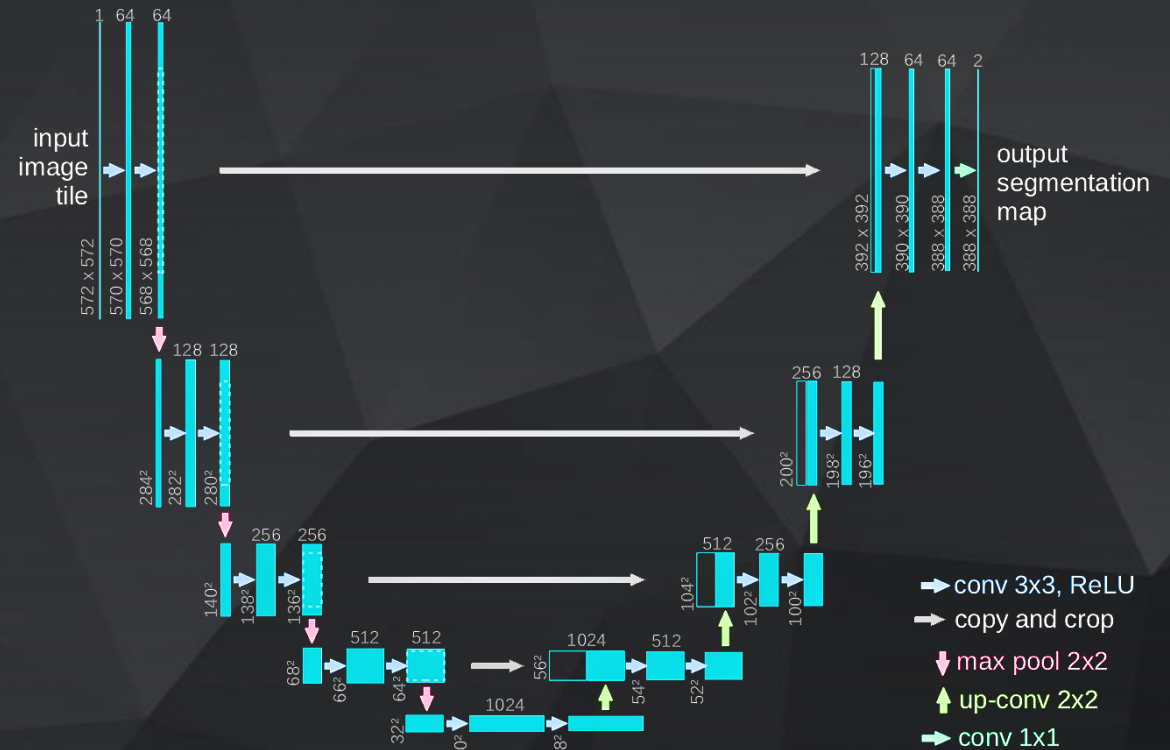


IMAGE + MASK TILE



B. TRAINING PHASE

- U-net is a convolutional network architecture for fast and high-accuracy segmentation of images.
- It's made of two paths: encoder, decoder
- **Encoder:** extracts features of different levels through a series of convolutions, ReLU activations and max poolings.
- **Decoder:** upsamples the result to increase its resolution of the detected features. . Skip connections are added to allow context and precise localization
- We also added batch normalization after each layer to improve speed, performance and stability.





C. POST-PROCESSING

- Submission file must contain all 11.481 single-band 1024x1024 TIFF files named after its corresponding imagery chip in the test set.
- Our output files are 512x512 binary masks with a 0-based format name.
- Having stored their real file names prior entering our prediction model, we proceed to restore them.

METRICS & LOSS FUNCTIONS

A metric is a function that is used to judge the performance of our model.

Average Accuracy: often used as a choice of evaluation for classification problems which are well balanced which means it's ideal for our binary classification problem.

$$\text{Accuracy} = (\text{Correct Predictions}) / (\text{Total Predictions})$$

The competition also makes use of the **Jaccard Index** metric as it's the main evaluation score for its participants. Jaccard is a similarity measure between two label sets and is defined as the intersection divided by the union.

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|}$$

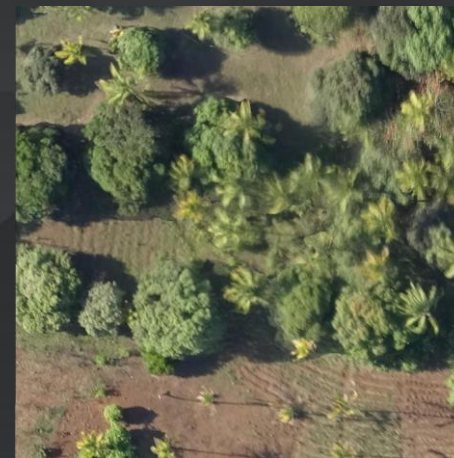
The loss function used to optimize our model is the Binary Crossentropy and it's used on true/false decisions. The Binary Crossentropy loss function measures the deviance of the true label from the predicted label.

$$L(\mathbf{y}, \hat{\mathbf{y}}) = -\frac{1}{N} \sum_{i=0}^N (y \cdot \log(\hat{y}_i) + (1 - y) \cdot \log(1 - \hat{y}_i))$$

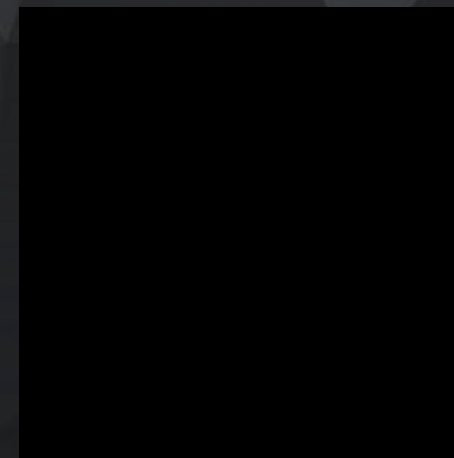
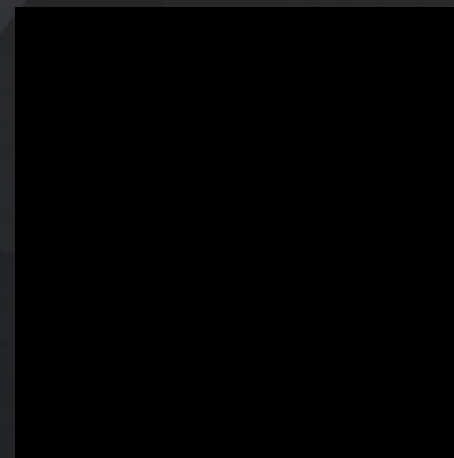
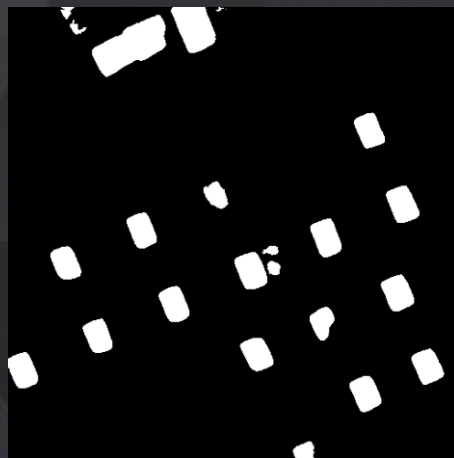
RESULTS

In the following images, we can see the results that we achieved. The first row represents the original images, while the bottom row are the generated masks by our model.

ORIGINAL
IMAGE



PREDICTED
MASK





REFERENCES

For more information, feel free to check out the included report.