BUILDING DETECTION USING OPTICAL IMAGE ANALYSIS

Team 6
CSYE 7200 - Term project

Shamin Choksi
Nina Shamsi
Chandrabhan Tanwar

Outline

- Introduction
- Use Cases
- Methodology
- Dataset source and content
- Milestones
- Acceptance Criteria
- Goals

Introduction

- The project aims to develop a deep learning model to accurately detect buildings in satellite images using GeoTrellis (Scala library), Spark and PyTorch.

- The project leverages the COCO and SpaceNet Rotterdam datasets, the latter of which is a collection of high-resolution satellite images of Rotterdam, Netherlands, with detailed annotations of building footprints.

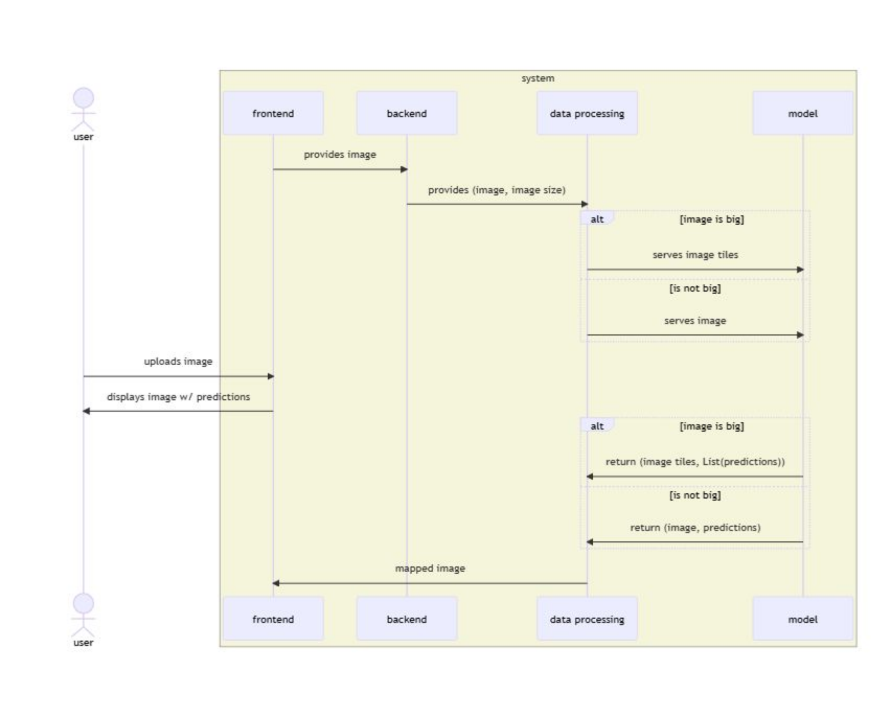
- Additionally, the project aims to optimize the model's performance by exploring various machine learning algorithms, hyperparameter tuning, and feature engineering.

Use Case - Predict Buildings

- User uploads a satellite image to the system

- System ingests the image and transforms it into a format recognisable by the model, which then returns the building data present in it

- System maps the results onto the image provided and returns the generated result back to the user



Methodology

The methodology for the project follows the standard for most deep learning pipelines and can be presented as a series of steps involving:

- Data ingestion and pre-processing
- Data augmentation and feature extraction
- Model training and hyperparameter tuning
- Evaluating model performance and deriving inference
- Deploying the model through an interface

Datasource and content: Rotterdam

As mentioned, one dataset we're using the SpaceNet AOI 11 – Rotterdam[1] dataset for training our model. Dataset description:

- Tiled geotiffs of 4-Band Multi-Spectral raster data from Maxar WorldView-2
- Tiled geotiffs of Panchromatic raster data from Maxar WorldView-2
- GeoJson labels of building footprints for each tile
- CSV of building footprint locations in pixel coordinates and orientation file indicating the directions from which each SAR image is captured (0 North, 1 South)

[1] Source: https://spacenet.ai/rotterdam/

Datasource: Rotterdam Images

- Multispectral images:
  - These are electro-optical RGB images in 4 bands (channels)
  - 4*16 bit, ~1m resolution
  - 450x450 px
- Panchromatic images:
  - These are electro-optical greyscale images in a single band (channel)
  - 16 bit, ~50cm resolution
  - 900x900 px

Image Source: https://spacenet.ai/rotterdam/

Datasource: Rotterdam Labels ~48,000


```json
{
    "type": "Feature",
    "geometry": {
        "type": "Polygon",
        "coordinates": [
            [
                [   4.359858006431244, 51.93177362131276],
                [   4.359881450267189, 51.9318702603761 ],
                [   4.360106255046586, 51.931853977422165 ],
                [   4.360082811547273, 51.93175733765214 ],
                [   4.359858006431244, 51.93177362131276 ]
            ]
        ]
    },
    "properties": {
        "img_id": "AOI_11_Rotterdam_PS-RGB_2019",
        "building_id": "1",
        "area": 5.948276306327426,
        "is_complete": true
    }
}
```


- GeoJson data:
  - Standard format for marking geography
  - Mostly will be used for building descriptors

- Summary Data:
  - This will act as the ground truth for the training data

  CSV file with the given format:

  [ImageId,TileBuildingId,PolygonWKT_Pix,Mean_Building_Height,Median_Building_Height,StdDev_Building_Height]

  <u>Example</u>: img339,1,"POLYGON ((213 269,184 221,130 249,154 293,162 288,165 292,169 290,180 285,213 269),(151 253,177 242,189 263,164 275,151 253))",1.75,1.75,0

Datasource: COCO (Common Objects in Context)

Provides bounding-box and per-instance segmentation

- We will use the buildings categories

Chu, Yi & Ahmadi, Hamed & Grace, David & Burns, David. (2021). Deep Learning Assisted Fixed Wireless Access Network Coverage Planning. IEEE Access. 9. 10.1109/ACCESS.2021.3108051.

Model

Segmentation model with pretrained encoder weights

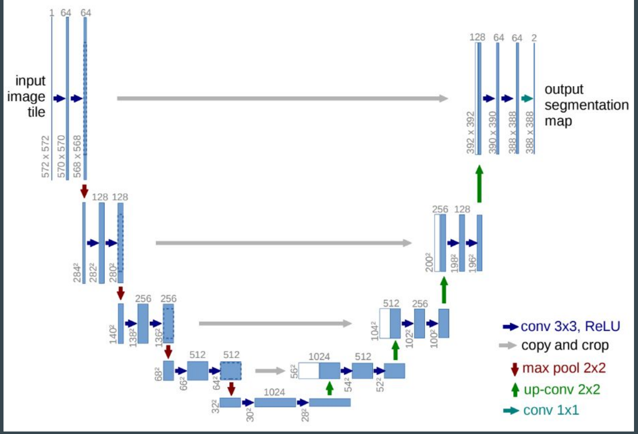
U-Net architecture

- ResNet50 encoder
- ImageNet encoder weights

Why we are using PyTorch: to lessen the resources

needed for training, we would like to use pretrained weights

from ImageNet for the ResNet50 encoder



Original architecture proposed by Ronneberger et. al (2015)
[1505.04597v1] U-Net: Convolutional Networks for Biomedical Image Segmentation (arxiv.org)

Milestones

- Data study, preparing environment and libraries
- Data pipeline, ingestion and pre-processing
- Feature extraction and data augmentation
- Model review, development and basic UI
- Model training and tuning
- Testing and preparing use case functionality
- Final touch-up

Acceptance Criteria

	Metric	Passing Criteria
Frontend	Response time	< 1ms
Backend	Data ingest	< 3 min (big) < 1 min (not big)
Data Processing	Imaging tiling	< 3 min (big) < 1 min (not big)
Model Training	IOU score Dice loss	>= 70 <= 0.5
Model Inference	Dice/IOU score	>=70

Goals

Develop a tool for:

- ingesting satellite image data,
- predicting building structures on a given image, and
- giving a prediction mask back to the user