Client Progress Review

ICT30001-Information Technology Project

V0.1

Prepared by :

Student ID	**Student ID**
Trac Duc Anh Luong	103488117
Minh Nghia Nguyen	103806269
Thanh Dat Nguyen	103804881
Anh Vu Pham	103806447
Tuan Nam Tran	103792643

1. Background & Introduction

NTT e-MOI JSC, a subsidiary of NTT East Japan, specialises in leveraging cutting-edge technology to drive business development and digital transformation for its clients. They offer various services including Managed Service Provider (MSP), IT Outsourcing (ITO), and global human resources training. In software development, they excel in using leading low-code platforms to deliver high-quality software products with shorter development times compared to traditional methods. NTT e-MOI strongly emphasises building long-term partnerships with customers by aligning with their business goals and fostering a collaborative, growth-oriented environment both internally and externally. (NTTe-MOI, 2020).

2. Project objectives

Goals

Given the situation of the client with the current inefficient, highly time and money-consuming process, the goal of this project is to create a highly efficient, highly available, online housing management system that not only provides accurate, detailed, and useful analysis but can also be accessed anywhere with an internet connection, at any time needed, while still maintaining high confidentiality, with strict access control that only allows verified, trusted individuals to access the system.

Objectives

The objective for this project can be divided into 2 parts. First, the image analysis model. This model will take in the current (most recent) satellite image (image taken from above) of the metro area and use that as a base. Other images of the same area but from the previous year(s) will then be inputted, and the model will compare the base images with its predecessors to find differences in terms of housing, land structure, landscape,... These data after being extracted will be transformed and moved to a storage medium - a database, for storage and further usage by the second part, the user interface application. This will be how the end users, our stakeholders will view the data generated by the analysis model (part 1). This will come in the form of a web application - accessed through the web. This ensures the application's availability, being able to be accessed from any device with a web browser and an Internet connection. This web application will use the data generated in part 1, and display it in both numerical and image, to give the user the most possible insights into the data, in a highly descriptive, but also easy-to-understand format. This website will also employ user authentication to ensure that only verified individuals with the correct credentials can access the application.

Expected benefits

This solution aims to give the stakeholder a more efficient, simpler, less expensive but also highly accurate and descriptive way to view changes in a metro area throughout the years.

Compared to the existing process, which requires a lot of man hours, manual labor and work, which is not only expensive, and inefficient but also highly error-prone, this new solution aims to improve it on all fronts, providing a highly accurate, insightful yet efficient and easy, convenient way to view these previous manually generated data, from anywhere, anytime. This solution will not only quicken up the process but also make it less error-prone, highly scalable, available, and more secure.

3. Scope

Analysing Housing Images from Urban Areas

The app will implement advanced computer vision techniques to analyse housing images from urban environments, prioritising Hanoi City. This process will involve:

- **Data Acquisition**: Collecting high-resolution images of houses from urban areas using various sources such as satellite images and drones from open sources like Kaggle and Google Earth.
- **Preprocessing**: The gathered data should be cleaned and preprocessed to guarantee quality and eliminate duplication.
- **Object Detection and Segmentation**: Identifying and isolating houses and other man-made architectural structures within the images. The steps might involve using convolutional neural networks (CNNs) and other deep learning models trained on the dataset we collected from the data acquisition step.
- **Change Detection**: Comparing new images with historical data to detect housing changes, such as new construction, renovations, and demolitions. This will help authorities identify and cross-check whether the construction is approved and licensed.

Implementing Smart Vision Algorithms Using AI Models

The app will incorporate advanced AI models to enhance the accuracy and performance of its vision algorithms. Key aspects will include:

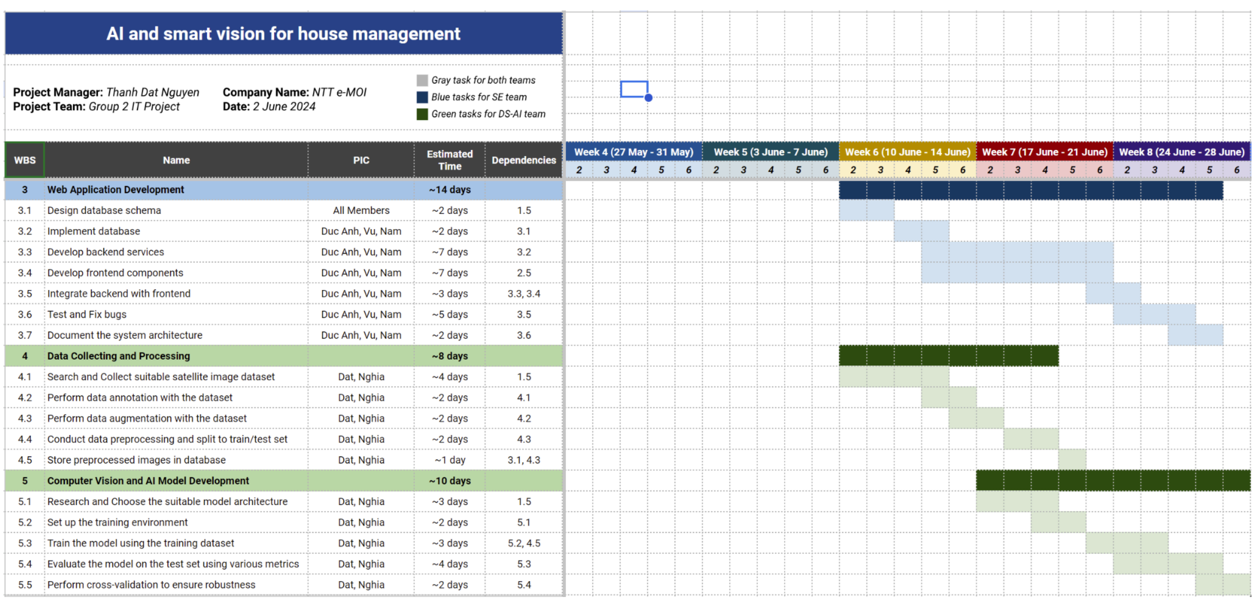
- **Machine Learning Models**: Utilising trained models and fine-tuning them with specific datasets to improve performance on ML tasks like object detection, image classification, and segmentation.
- **Deep Learning Frameworks**: Implementing frameworks like TensorFlow, PyTorch, or Keras to build and deploy the AI models. These existing frameworks are essential to creating neural networks that can process and analyze large volumes of image data.
- **Continuous Learning**: As more images are analyzed, the AI model will update and refine its understanding, aiming for higher accuracy over time.

User Interface with 2D Map Displaying House Properties

Users will be able to visualize housing data with the informative and user-friendly design of the interface. Important features will include:

- **Interactive 2D Map**: A user-friendly map that displays the distribution of houses in selected urban areas. Users can zoom in and out, pan across different neighborhoods, and click on the house pin for more details.
- **Property Information**: Each house on the map will be annotated with key properties like:
    - Address
    - Latitude and Longitude
    - Area
    - House Owner
    - Yearly Changes in the Area
- **Search and Filter Options**: Tools that let users narrow down their results by size, ownership, or historical changes or search for particular properties. As a result, users will find the information they need more quickly.
- **Data Export**: The user can export property data and map visuals for download and sharing with standard formats like CSV, PDF, or image files.

4. Progress to date



WBS	Name	PIC	Estimated Time	Dependencies
3	**Web Application Development**		~14 days	
3.1	Design database schema	All Members	~2 days	1.5
3.2	Implement database	Duc Anh, Vu, Nam	~2 days	3.1
3.3	Develop backend services	Duc Anh, Vu, Nam	~7 days	3.2
3.4	Develop frontend components	Duc Anh, Vu, Nam	~7 days	2.5
3.5	Integrate backend with frontend	Duc Anh, Vu, Nam	~3 days	3.3, 3.4
3.6	Test and Fix bugs	Duc Anh, Vu, Nam	~5 days	3.5
3.7	Document the system architecture	Duc Anh, Vu, Nam	~2 days	3.6
4	**Data Collecting and Processing**		~8 days	
4.1	Search and Collect suitable satellite image dataset	Dat, Nghia	~4 days	1.5
4.2	Perform data annotation with the dataset	Dat, Nghia	~2 days	4.1
4.3	Perform data augmentation with the dataset	Dat, Nghia	~2 days	4.2
4.4	Conduct data preprocessing and split to train/test set	Dat, Nghia	~2 days	4.3
4.5	Store preprocessed images in database	Dat, Nghia	~1 day	3.1, 4.3
5	**Computer Vision and AI Model Development**		~10 days	
5.1	Research and Choose the suitable model architecture	Dat, Nghia	~3 days	1.5
5.2	Set up the training environment	Dat, Nghia	~2 days	5.1
5.3	Train the model using the training dataset	Dat, Nghia	~3 days	5.2, 4.5
5.4	Evaluate the model on the test set using various metrics	Dat, Nghia	~4 days	5.3
5.5	Perform cross-validation to ensure robustness	Dat, Nghia	~2 days	5.4

As outlined in the pre-established Gantt chart, the project has successfully completed both the website development and the data collection and processing phases. This milestone marks significant progress in our overall project timeline.

During the website development phase, our team designed, built, and tested a fully functional platform that meets all specified requirements. The web demo now offers users the ability to view each independent house with clear indicators and key information upon clicking. Additionally, the website includes robust functionalities for data upload and visualization. At the moment, the dataset encompasses a diverse array of images from various areas around Hanoi, providing a comprehensive visual representation of the region's housing. The website is now live and operational, providing a user-friendly interface and robust back-end functionalities.

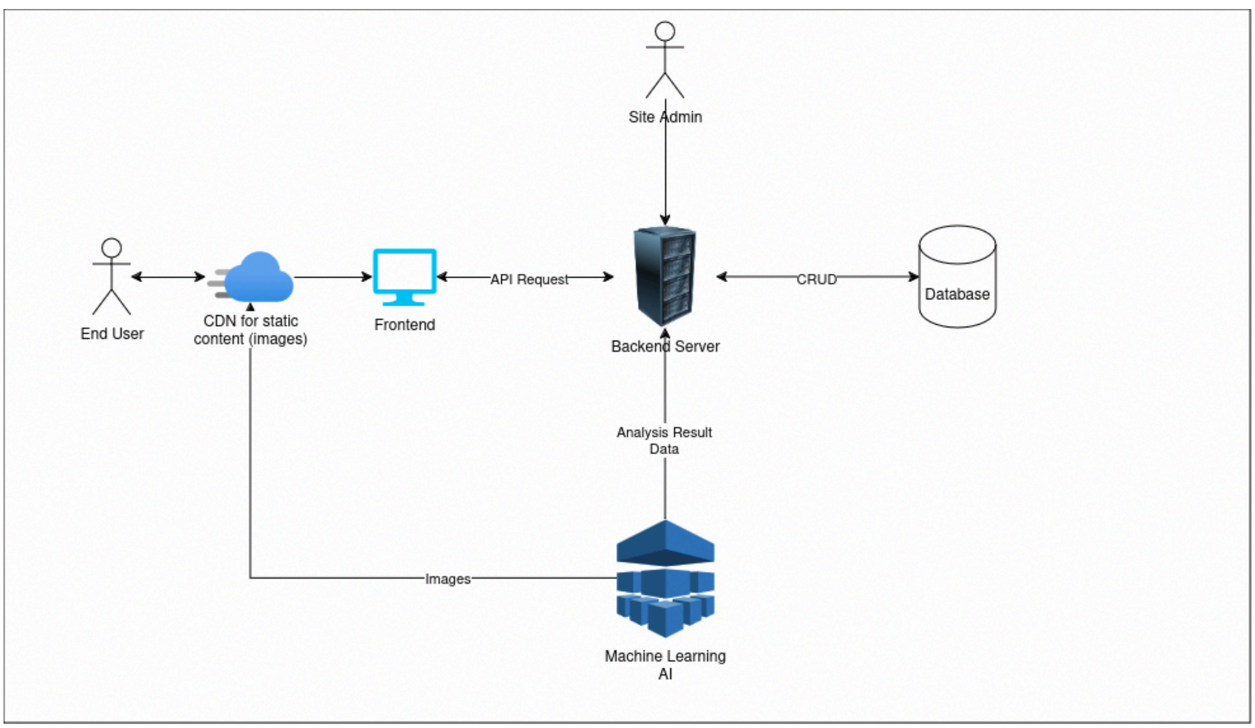
Upon the completion of the website, we simultaneously reach the final stages of data collection and processing. This involves an integrated approach where we initially gather house images from various online sources. Following this, we cleaned and augmented the dataset to ensure high quality and consistency. Once the dataset is refined, we upload it to the database, making it fully prepared for AI model training.

With these critical phases now behind us, we are well-positioned to move forward with the subsequent stages of the project. The next steps will involve developing the AI model, deployment and integration of the model, and project documentation.

5. Progress so far

SE Team

Architecture design

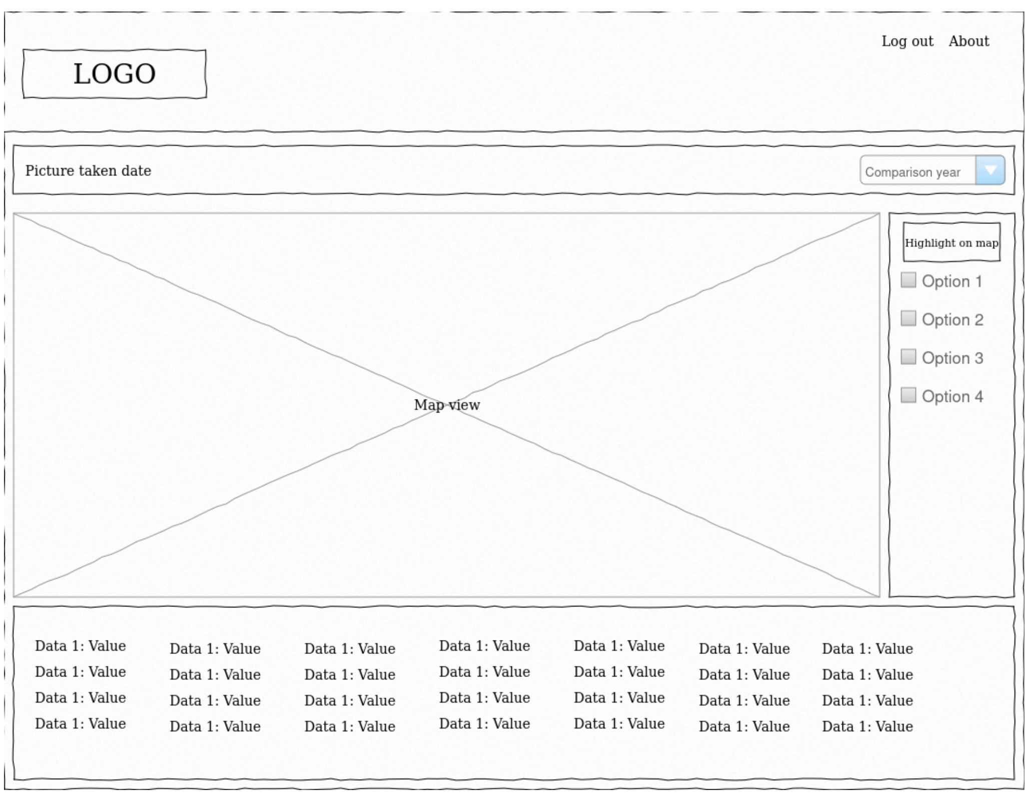


For the architecture, we have come up with this design, which includes:

- **Backend server**: This will host all of our behind-the-scenes logic involving the querying and managing of data. This will be accessible to the front end, the Machine Learning server as well as a site admin through SSH. This backend will also be the only way to directly interact with the application's data
- **The Database**: Where the application's data will be stored. Only accessible through the backend server
- **Machine Learning AI**: A separate server to run the machine learning model to generate analysis of the images. The data generated from this process will be sent to the backend server through an API, and the backend server will process this data before storing it in the database
- **Frontend**: This server will do the job of getting the data required from the backend, then drawing it in a UI that will be displayed to the end user
- **CDN**: Where the images and cached content will be stored. This ensures low latency for end users, as well as to protect the main application from possible attacks

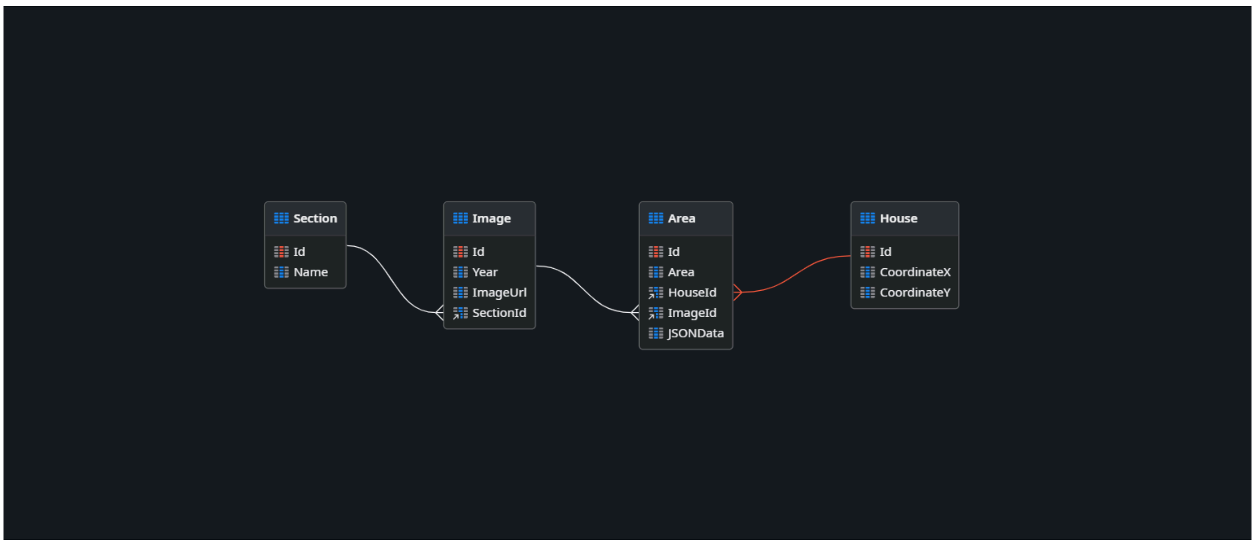
Wireframe design

This task's goal is to create a wireframe (i.e. a rough sketch of what the UI will look like). This wireframe focuses on the layout and functionalities of the UI and will not delve deep into the actual design (color schemes, themes,...)



For this application, due to the focus being mainly on simplicity and clarity of features and data above all else, as well as the time constraint for the actual development, we have gone with a simple UI design. The main feature that stands out is going to be the map view. This is the map of the area being analyzed as of the most recent satellite image. The date of this image will be displayed on the top left. The user can choose on the right which previous year to run comparisons on. This process will generate data comparisons between the current image of the area with the image from previous years. Data and the differences will be displayed at the bottom of the web page. Furthermore, the user can choose to highlight details on the map. They will be allowed to enable any/all highlights on the map as per their liking. On the top of the page will be the logo, the About page, as well as the option to log out of the website.

Database design



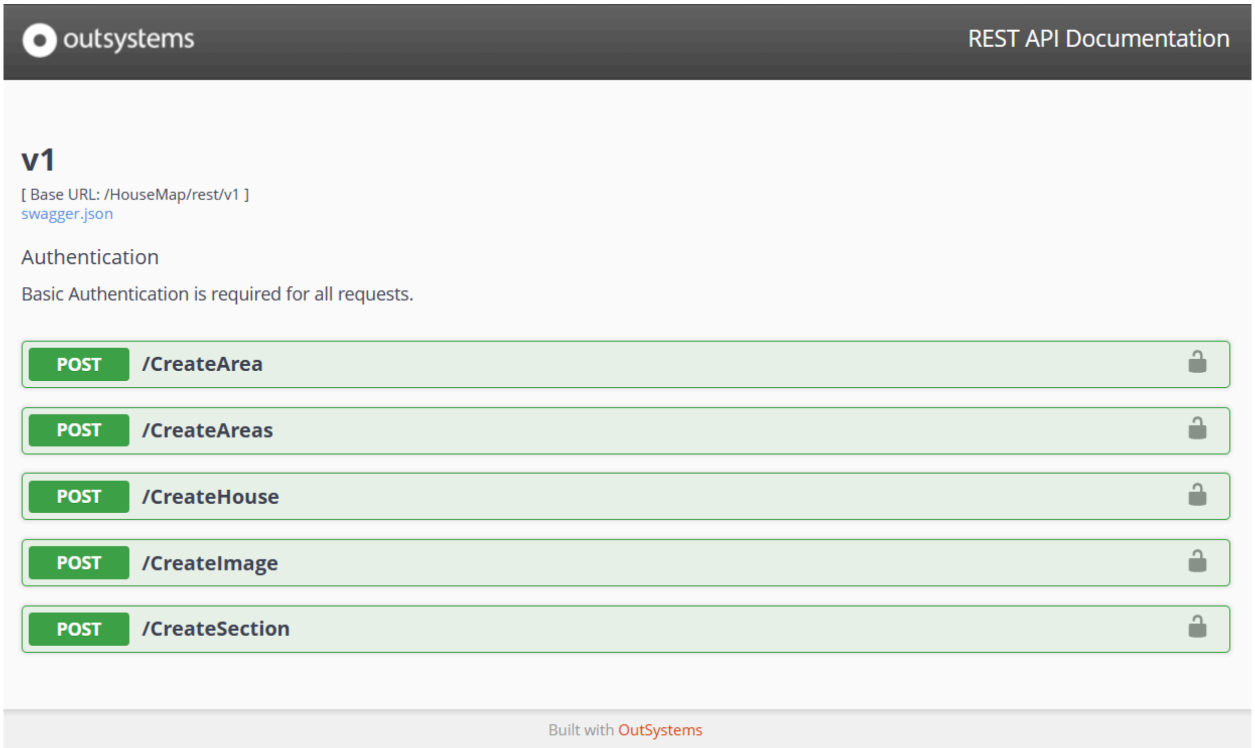
Database schema

The database will include 4 tables:

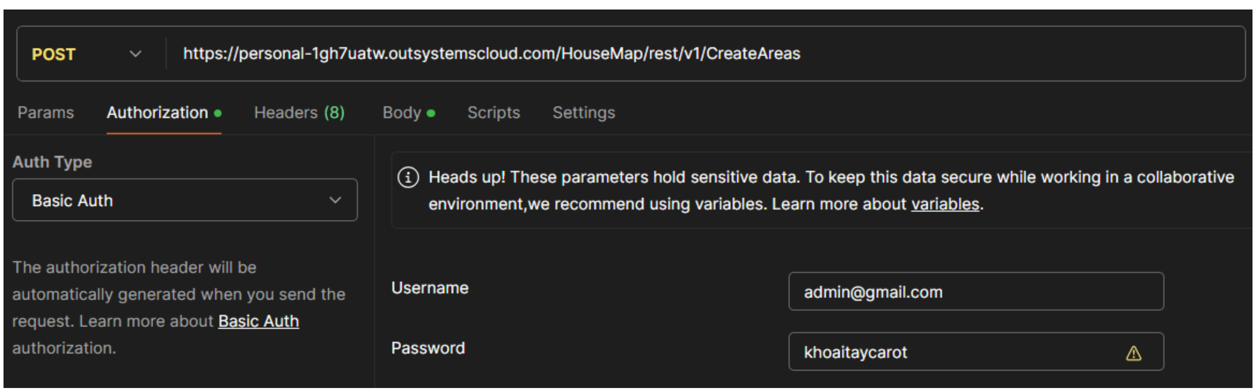
- Section: Stores the section of the map
- Image: The information of the images that can be used for displaying
- Area: Stores all information about the areas of houses in the pictures
- House: Stores information about individual houses (which image they belong to, coordinates,...)

APIs for DS model

Link to swagger: https://personal-1gh7uatw.outsystemscloud.com/HouseMap/rest/v1/

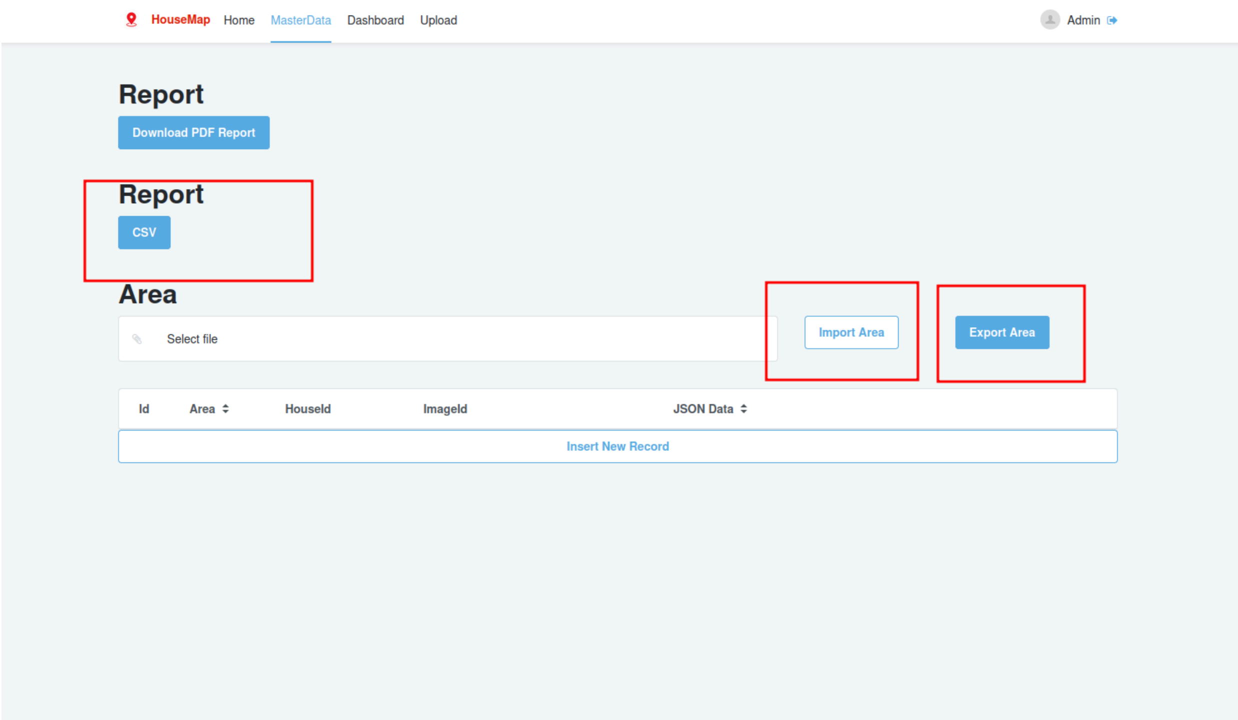


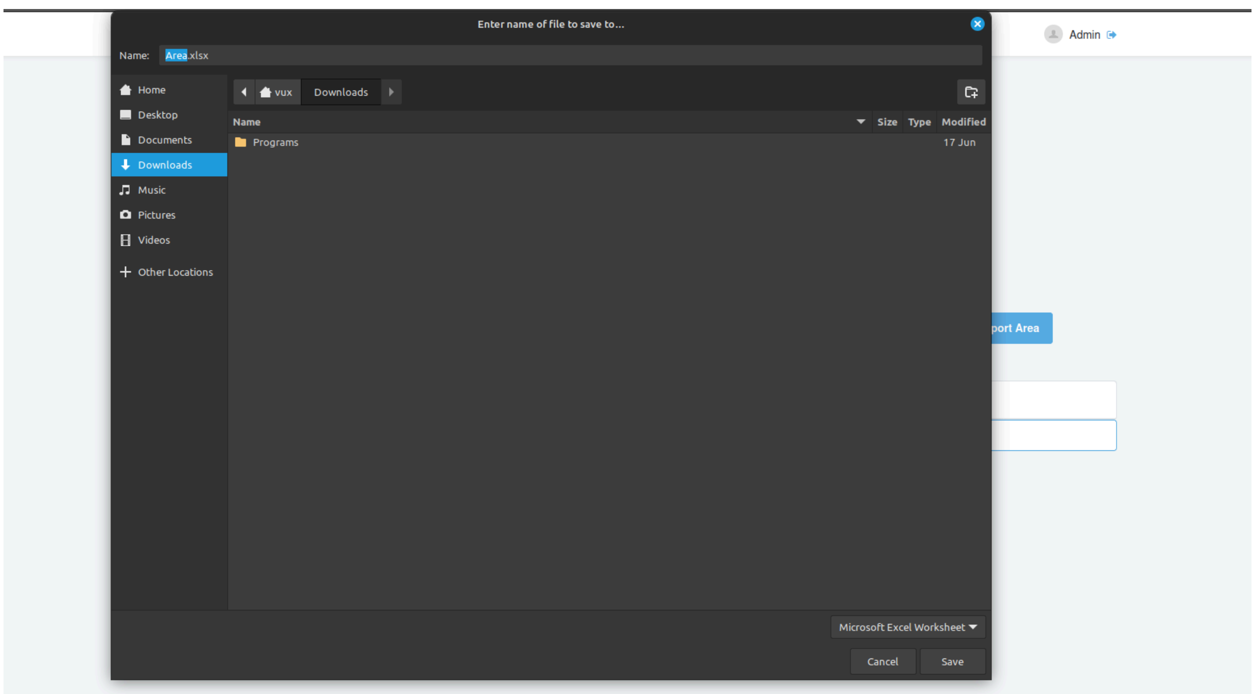
Authorization with **Basic Auth** **:**

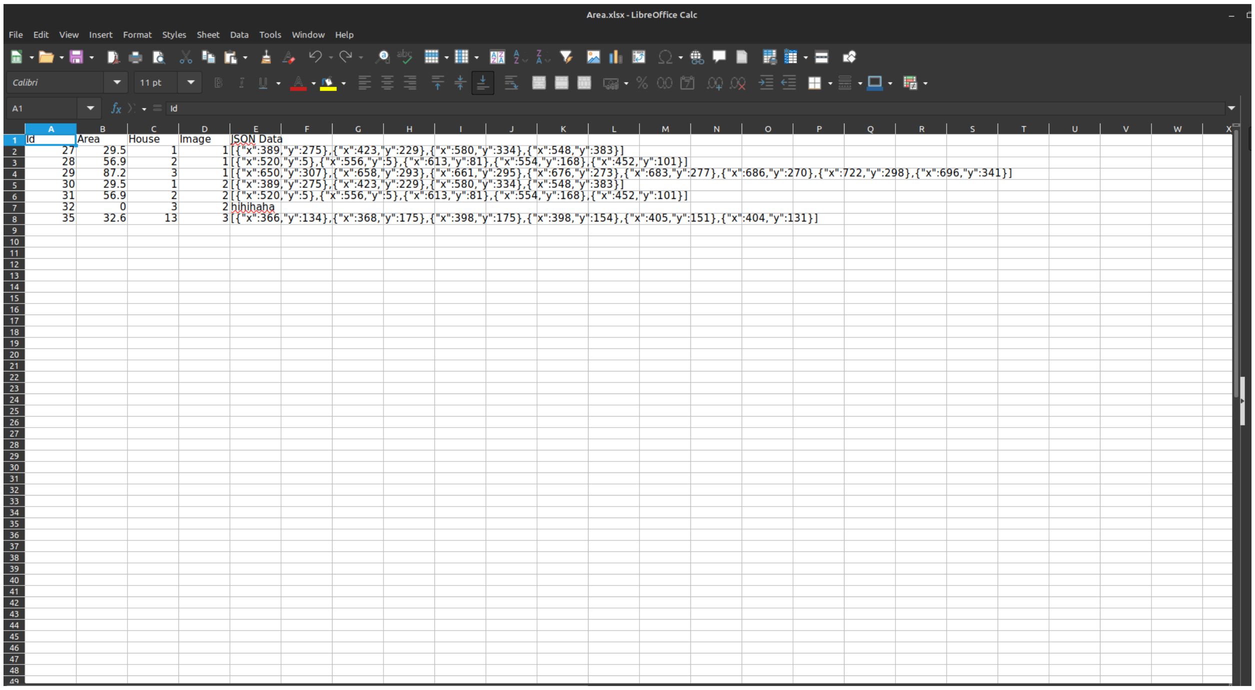


We have created an API endpoint for the application to inject data to be displayed. This API will be used by the DS team to inject data generated from the Model's Analysis into the Application's database. These data will be used to draw up / display the UI for the end user

Import / Export data as Excel sheets

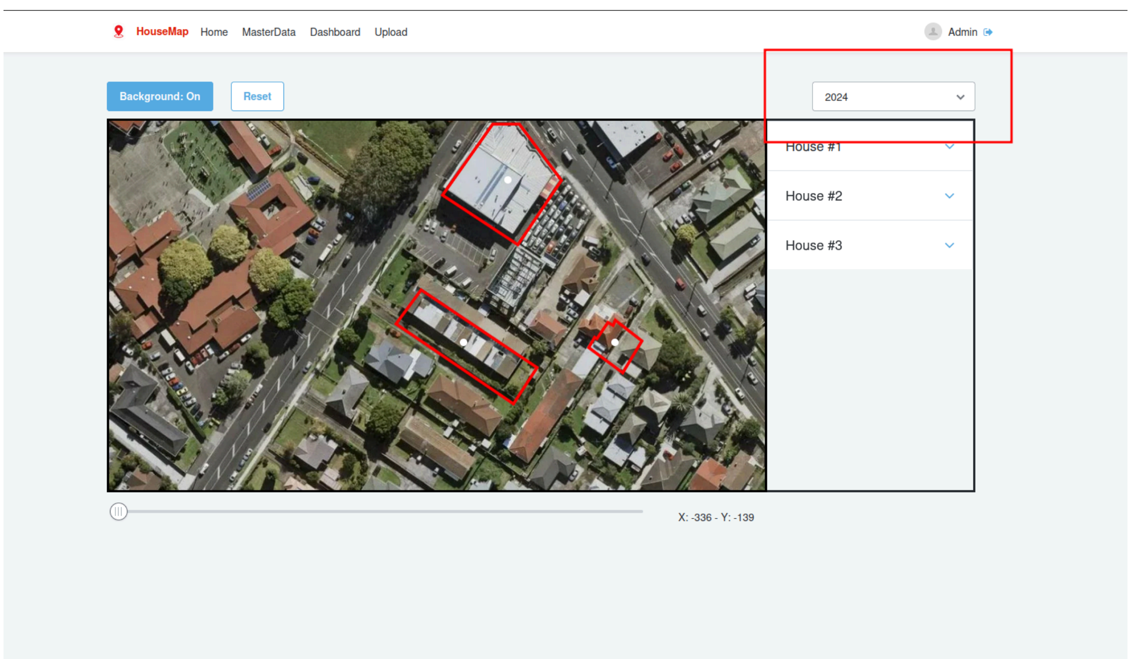

HouseMap
Home
MasterData
Dashboard
Upload
Admin
Report
Download PDF Report
Report
CSV
Area
Select file
Import Area
Export Area
Id
Area
HouseId
ImageId
JSON Data
Insert New Record


Enter name of file to save to…
Name: Area.xlsx
Home
Desktop
Documents
Downloads
Music
Pictures
Videos
Other Locations
vux
Downloads
Name
Size
Type
Modified
Programs
17 Jun
Microsoft Excel Worksheet
Cancel
Save
Admin



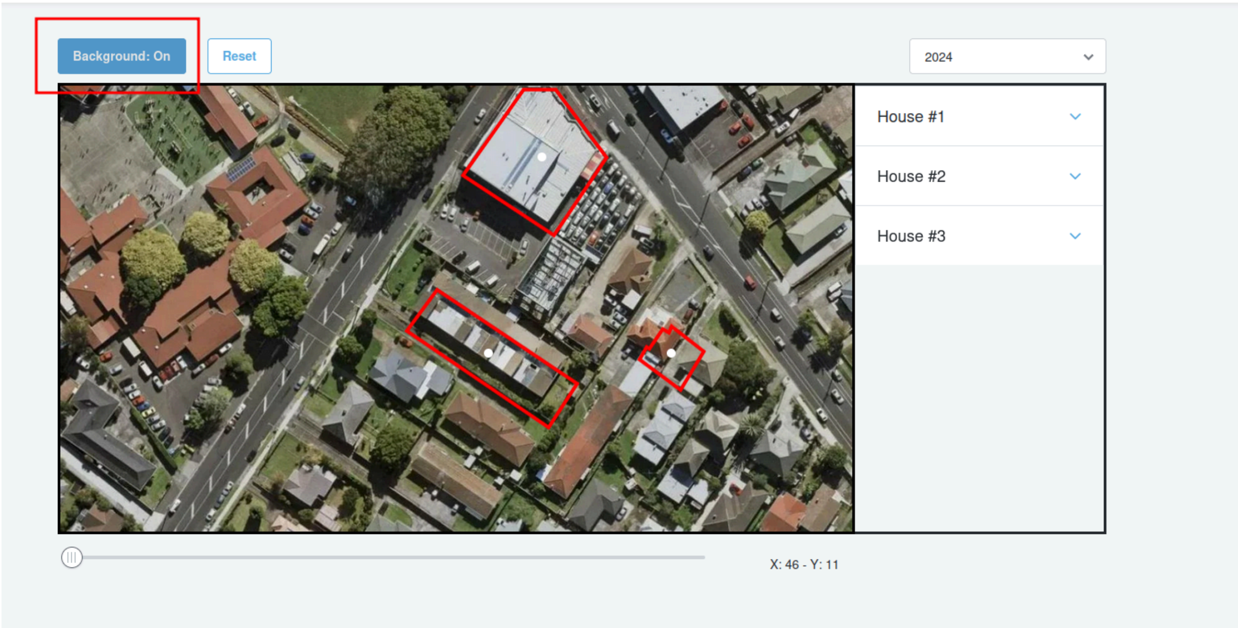
We have also added the ability for admins to import their data, as well as export the data in the application in the form of Excel spreadsheets.

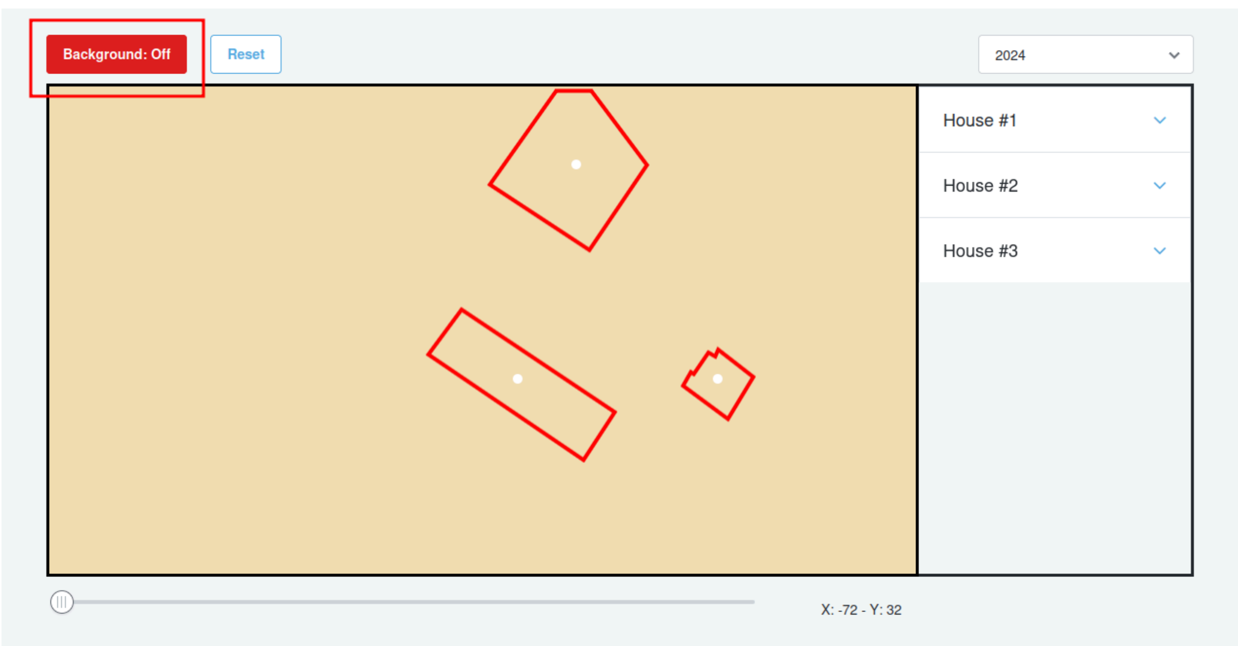
Year picker for Web UI



We have also added the option to specifically choose the year for the image in the UI. This directly references the records in the database.

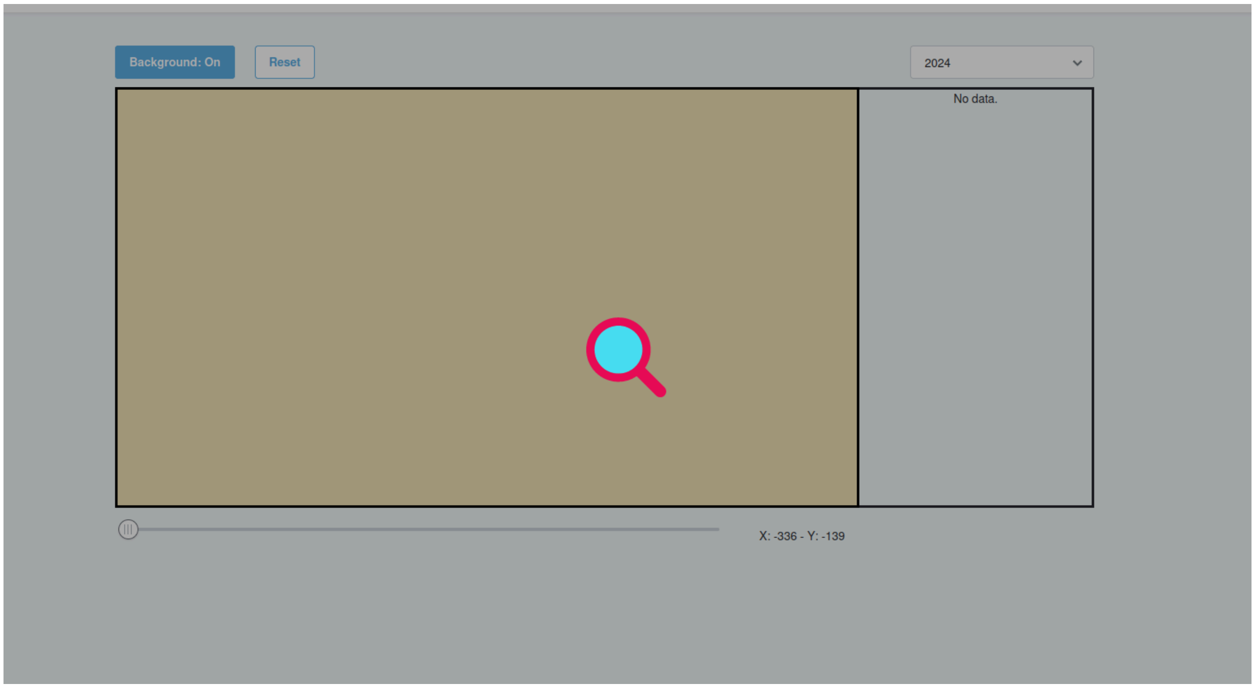
Option to toggle background





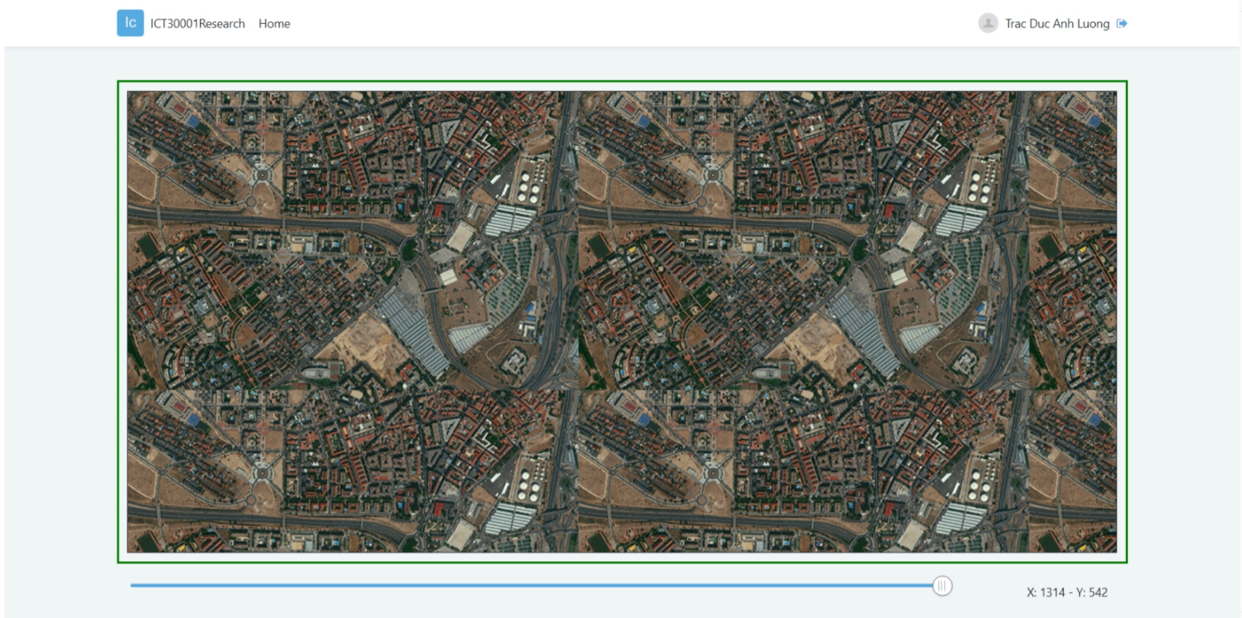
Another option added was the ability to toggle the background image. This allows the user the option to remove the actual image to get a better view of the model-defined areas of housing, without the clutter.

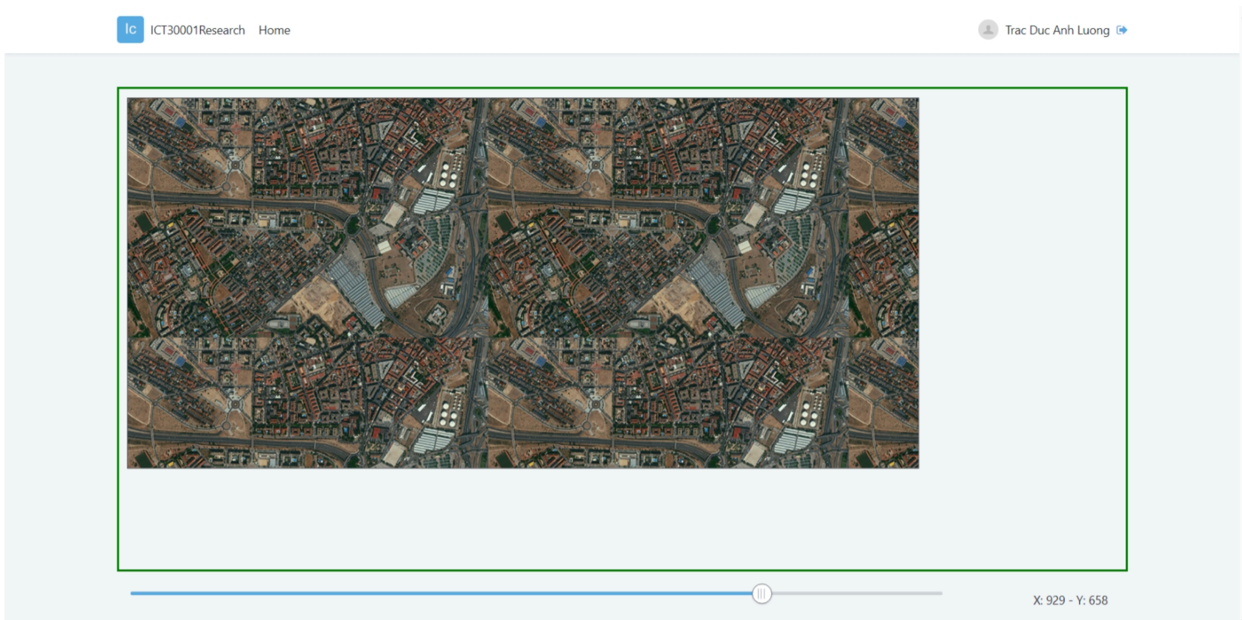
Loading animation



To offset the long initial load time of the images, which will happen as the images are quite detailed at a high resolution (8K), we have also added a loading animation, which will run as a placeholder when we are loading the images into the UI.

Research Implementing Coordination Mapping to Map Images





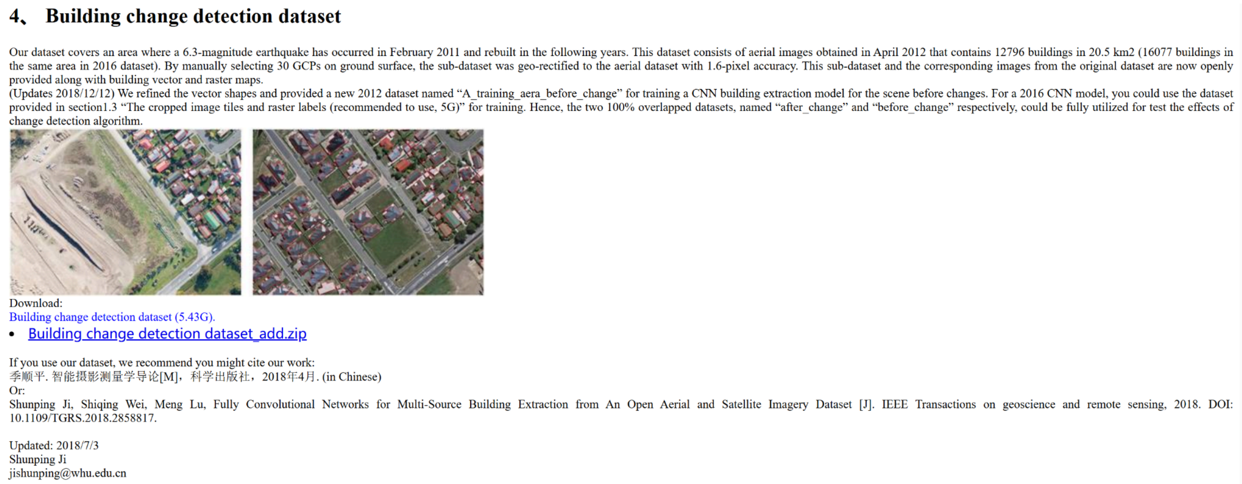
Sample: https://personal-1gh7uatw.outsystemscloud.com/ICT30001Research/

DS Team

Research for satellite image dataset

The most important thing to start to build an AI model is to have a good dataset for training and testing it. After conducting research, we found the WHU building dataset as a suitable choice. This dataset, created by a team from WHU, covers 20 km² in Christchurch, New Zealand, using various remote sensing resources with high resolution, making it ideal for our model.

Link: gpcv.whu.edu.cn/data/building_dataset.html



4、 Building change detection dataset

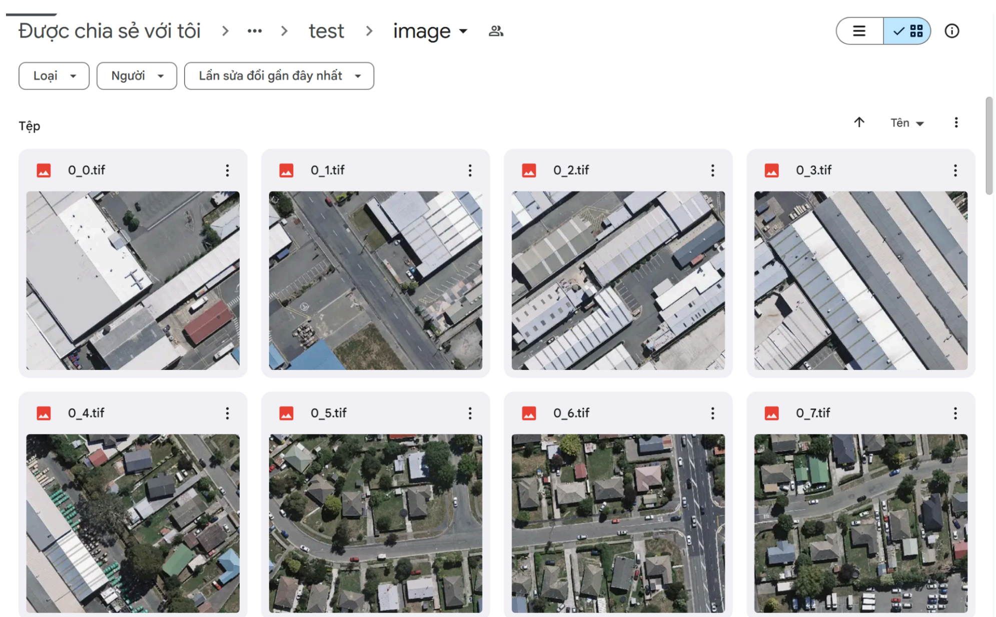
Our dataset covers an area where a 6.3-magnitude earthquake has occurred in February 2011 and rebuilt in the following years. This dataset consists of aerial images obtained in April 2012 that contains 12796 buildings in 20.5 km2 (16077 buildings in the same area in 2016 dataset). By manually selecting 30 GCPs on ground surface, the sub-dataset was geo-rectified to the aerial dataset with 1.6-pixel accuracy. This sub-dataset and the corresponding images from the original dataset are now openly provided along with building vector and raster maps.
(Updates 2018/12/12) We refined the vector shapes and provided a new 2012 dataset named “A_training_aera_before_change” for training a CNN building extraction model for the scene before changes. For a 2016 CNN model, you could use the dataset provided in section1.3 “The cropped image tiles and raster labels (recommended to use, 5G)” for training. Hence, the two 100% overlapped datasets, named “after_change” and “before_change” respectively, could be fully utilized for test the effects of change detection algorithm.

Download:
Building change detection dataset (5.43G).
- Building change detection dataset_add.zip

If you use our dataset, we recommend you might cite our work:
季顺平. 智能摄影测量学导论[M]，科学出版社，2018年4月. (in Chinese)
Or:
Shunping Ji, Shiqing Wei, Meng Lu, Fully Convolutional Networks for Multi-Source Building Extraction from An Open Aerial and Satellite Imagery Dataset [J]. IEEE Transactions on geoscience and remote sensing, 2018. DOI: 10.1109/TGRS.2018.2858817.

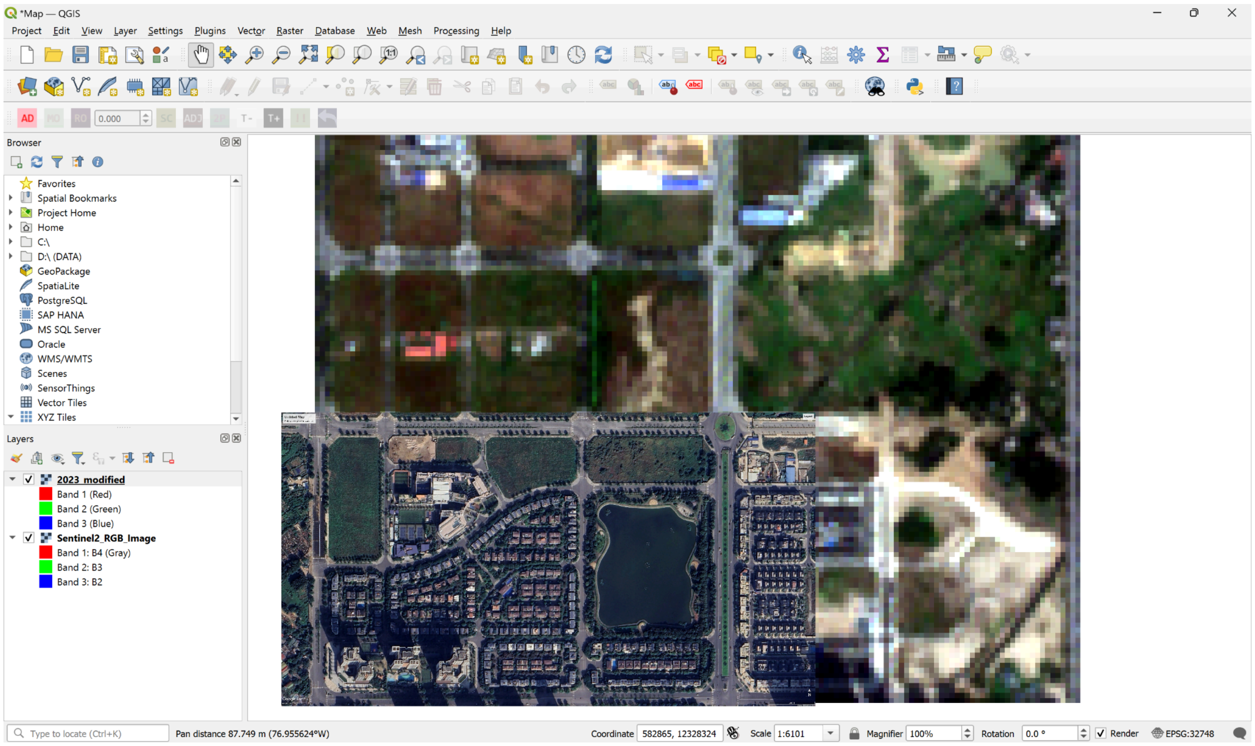
Updated: 2018/7/3
Shunping Ji
jishunping@whu.edu.cn

We have downloaded the dataset and uploaded it to the shared drive for easier access. The data is well-labeled, in TIF format, and already split into 512x512 images. It is organized into two folders, 2012 and 2016, containing images of the same area from these two years.



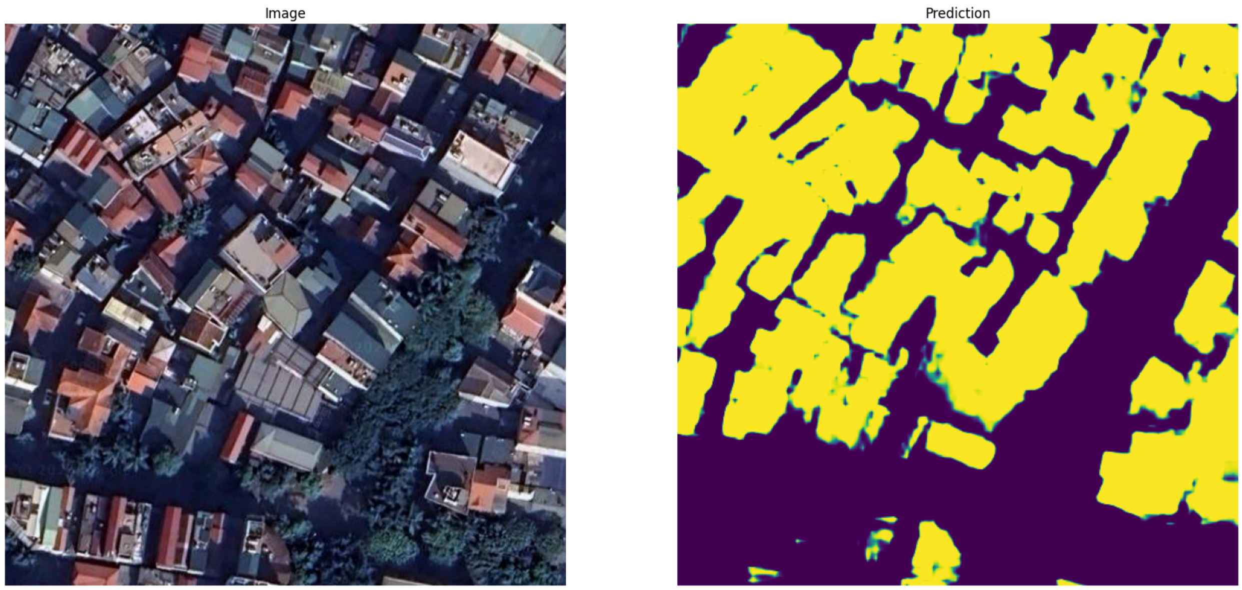
Production Data source

For this project, we need to segment and calculate the building area of different years in Hanoi using satellite images. To fulfill this requirement, we've chosen to use Google Earth Engine (GEE) for its free access and high accuracy in exporting TIF files, and Google Earth Pro (GEP) for exporting high-resolution images (1k to 8k). We exported a GEO_TIF file from GEE for our area of interest and then loaded it into QGIS, a GIS tool for georeferencing, to map the high-resolution image with the true coordinates on the GEO_TIF file from GEE. The final output will be a TIFF file with high resolution and mapped to the true coordinate on the map.

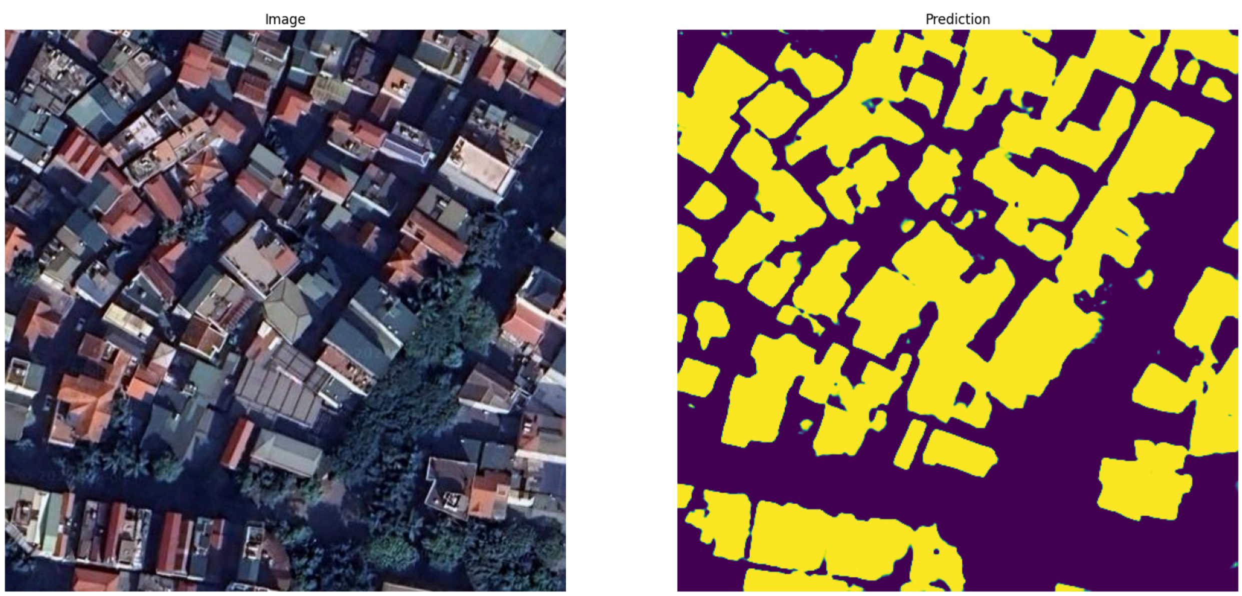


Train and Evaluate the model for housing segmentation

We trained the model using the WHU building dataset and tested it on a random area in Hanoi. During the first week, the model's output was not as accurate as expected; it struggled to distinguish between two separate houses in the segmentation process.



In the following week, we retrained the model for more epochs. This resulted in some improvements in the output, with the model better distinguishing between separate houses in the segmentation process.



Map the coordinate of the building with the real coordinate

Using the TIF file that has been georeferenced using QGIS, we can based on that and detect the coordinate of the house we want, ensuring that each segmented building accurately corresponds to its true geographical location.

Image processing complete!

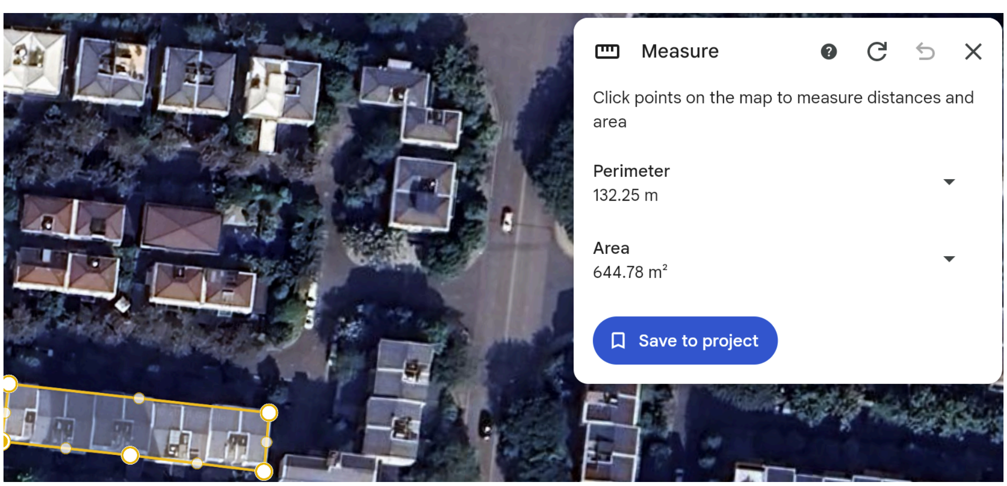
```
[
  0 : {
    "area" : 643.715269156804
    "shape" :
      [ 0 - 100 ]
      [ 100 - 103 ]
    "center" : {
      "x" : 1500
      "y" : 993
    }
    "building_id" : 23
    "coord" : {
      "lat" : 21.052102562116673
      "lon" : 105.79422842381483
    }
  }
```


Calculate the house area from the segmentation

Next, based on the coordinates of the houses and the model's segmentation results, we calculated the house areas in the images. The accuracy of these calculations was quite close to the area measurements obtained from Google Earth.

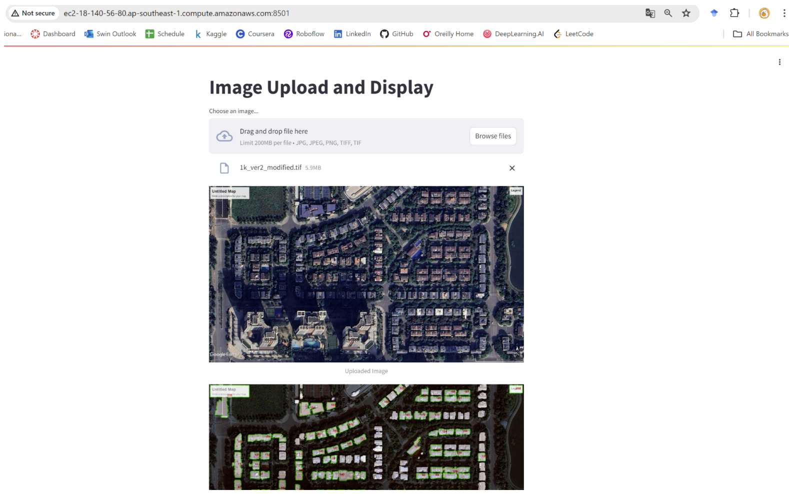
Image processing complete!

```
[
  0 : {
    "area" : 643.715269156804
    "shape" :
      [ 0 - 100 ]
      [ 100 - 103 ]
    "center" : {
      "x" : 1500
      "y" : 993
    }
    "building_id" : 23
    "coord" : {
      "lat" : 21.052102562116673
      "lon" : 105.79422842381483
    }
  }
```

Deploy the pipeline on Cloud AWS for demonstration

We have deployed the trained model on an EC2 instance and implemented an interface for uploading and testing the model segmentation and house area calculation for real input data.



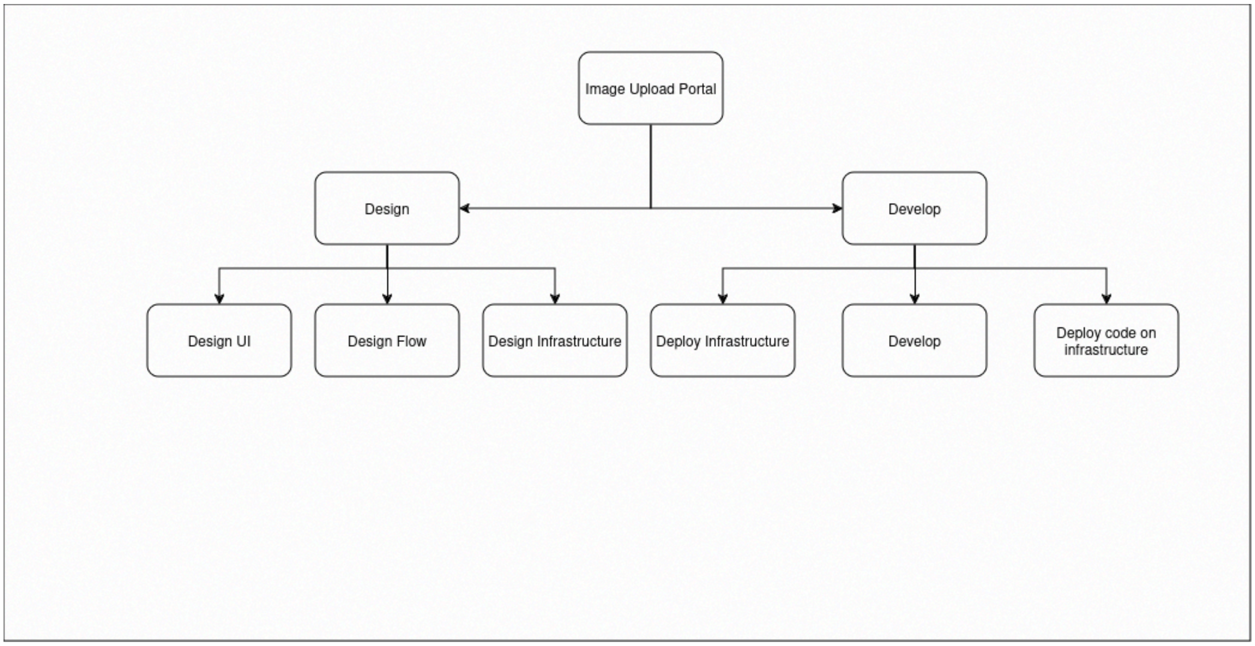
6. Future tasks

SE Team

For the next step of the project, we will research the position mapping technique between satellite image position and mouse cursor position on canvas. This research is instrumental in ensuring precise interaction and navigation within the proposed application. Our focus will be on developing a method to accurately translate the mouse position to the corresponding location on the satellite image, taking into account relevant information such as image resolution, canvas dimensions, and user interactions.

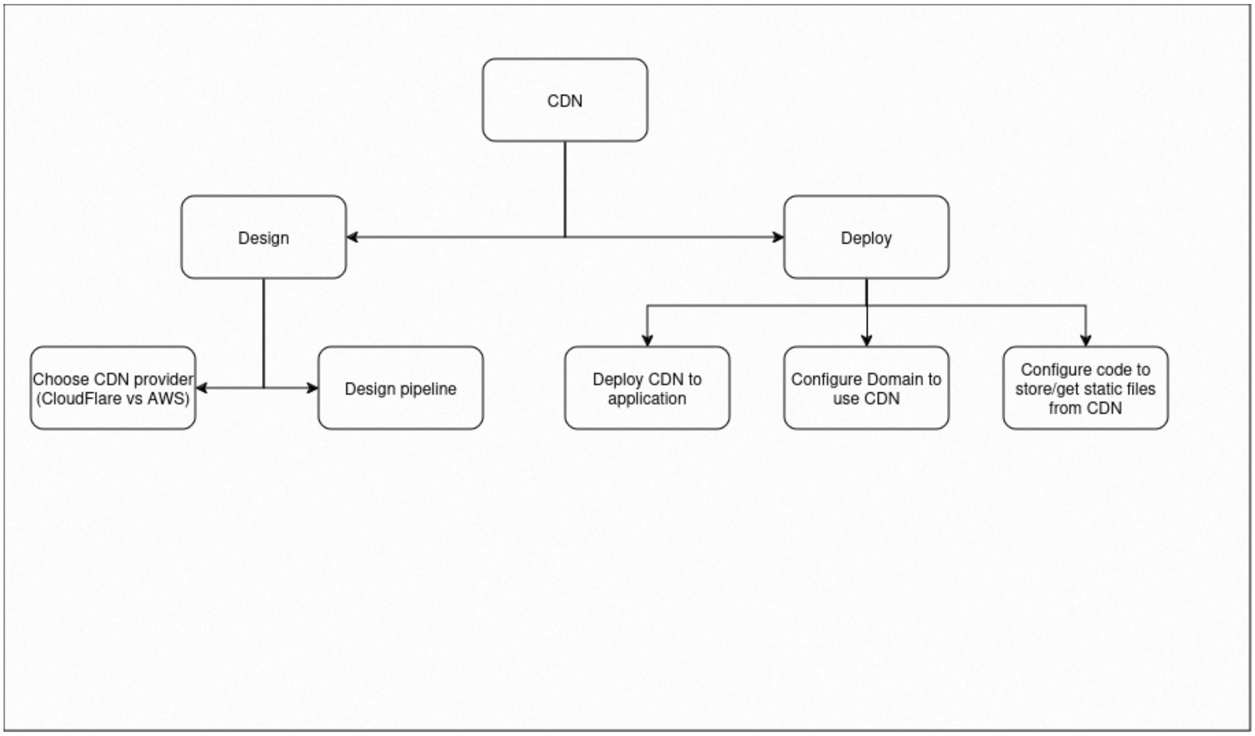
There is also the requirement for sizing and ratio mapping as we would want the user to be able to scale the images (zoom in, zoom out). We will explore different techniques to maintain image clarity and detail at various zoom levels. By effectively implementing sizing and ratio mapping, the user experience can be significantly enhanced with intuitive and responsive controls.

Image upload portal



Next, we also plan to create an image upload portal for users. Currently, we are using pre-chosen images of areas for the application. This step will allow users to upload their own images (.jpg, .png, .tif, .tiff), which will then be analyzed by the model to generate analysis. These analysis will be loaded to the database, which will allow these user-uploaded images to be used in the application just like the pre-chosen images

Setup CDN



Another step we are going to achieve is to apply a CDN (Content delivery network) to the application. This includes moving the images to the CDN, as well as route the CDN to the actual frontend application. This allows for faster load time for both the web app and the images, as well as ensures security provided by the CDN provider. We also need to consider our choices, which currently include CloudFlare and AWS

Testing

Finally, for this stage, we also need to write test cases for the environment and conduct those tests. These test cases will include:

- Logic testing
- Infrastructure testing
- Usability testing

This process will help identify errors and problems that may have been overlooked during the development process

DS Team

Improve the model accuracy

We aim to test out more architecture combinations, including models such as Mix Vision Transformer and MANet. By experimenting with these different architectures, we hope to identify which one achieves the best results for our project. This step is crucial to optimize our model's performance and ensure we are using the most effective approach.

Extract and georeference more areas of Hanoi

As for now, we have only exported yearly images for a specific area in Hanoi (specifically, khu đô thị hồ Tây) for testing purposes. Moving forward, we plan to expand our mapping efforts to cover more areas of Hanoi and integrate them into the model for further analysis and application.

Create API for integrating with the Web Application

One of the crucial goals is to build and expose an API that allows the FE to trigger whenever users (i.e. admins) upload new section images. This would then call back to the backend API to trigger the inference process and then push data to the database for display on the FE side.

7. Deliverables

AI and Smart Vision House Management App

- **Functionality**: Develop an application that uses AI and smart vision technology to track house changes using satellite images.
- **Platforms**: Ensure the app is accessible on web platforms for maximum convenience.

Smart Vision System

- **Development**: Develop a Smart vision system specifically designed for detecting property changes using satellite imagery. This system could encompass multiple Machine Learning models, which would be handled as part of the deliverables.
- **Performance**: Optimize the system for accuracy and efficiency, ensuring minimal false positives and negatives.

Training Data

- **Data Collection Pipelines**: A comprehensive documentation of data pipelines that were developed to collect, process, and store data used for training the Smart Vision system.

System Design Documentation

- **Architecture Design**: Provide detailed documentation of the system architecture, including the integration of AI components, data sources, and user interfaces.

- **Technical Specifications**: Document the technical specifications, including hardware and software requirements, data flow diagrams, and component interactions.

Code Documentation

- **Source Code**: Deliver the complete source code of the application, including AI models and integration components.
- **Code Comments**: Ensure the code is thoroughly commented on to explain the functionality and logic of key sections.
- **API Documentation**: Provide detailed documentation for any APIs developed as part of the project, including usage examples and endpoint descriptions.

User Training and Documentation

- **User Manuals and Guides**: Provide comprehensive documentation, including user manuals and quick-start guides, to assist users in navigating the app.

System Integration and Testing

- **Integration**: Ensure seamless integration of the app with existing NTT e-MOI systems and any third-party tools required by clients.
- **Testing**: Conduct extensive testing, including unit tests, integration tests, and user acceptance testing (UAT), to ensure the app's reliability and accuracy.