Final Report - AI and smart vision for house management

ICT30001-Information Technology Project

Client: NTT e-MOI JSC

Prepared by :

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

TABLE OF CONTENTS

Executive Summary

NTT e-MOI JSC, a subsidiary of NTT East Japan, is leading the way in digital transformation with its innovative Managed Service Provider (MSP), IT Outsourcing (ITO), and global human resources training solutions. Their latest initiative aims to transform property management by implementing a sophisticated AI and Smart Vision system, addressing the inefficiencies of the current manual property inspection methods, which are time-consuming and prone to errors. The new system is designed to replace these outdated processes with an automated solution that significantly enhances accuracy and operational efficiency.

The main objective is to create an advanced, highly accessible online housing management system that delivers precise and detailed analyses of housing changes over time. By leveraging the latest image analysis technology, the system will compare current satellite images with historical data to quickly identify structural changes. The collected data will be securely stored and made available through an advanced web application, ensuring high confidentiality and ease of access for authorized users.

This solution offers numerous benefits. In the short term, it will greatly reduce inspection times and operational costs while improving monitoring reliability. In the long term, it will position NTT e-MOI as a leader in innovative property management solutions, boosting customer satisfaction and competitive edge.

The web application was developed using OutSystems, a low-code platform chosen for its flexibility, performance, and familiarity to the development team. The application includes features for data visualization, interactive mapping, and secure data handling, providing users with comprehensive tools for property management. The AI component employs a Mix Vision-Transformer and MANet architecture to analyze housing images. Despite some limitations in speed and image quality, the solution achieves high accuracy and incurs no additional costs, making it suitable within the given constraints.

In conclusion, the AI and Smart Vision for House Management project successfully addresses the client's needs for a more efficient, accurate, and scalable property management system. By leveraging advanced AI and vision technologies, the solution offers significant improvements over the current manual processes. Ongoing enhancements and user feedback will ensure the system remains effective and relevant, providing long-term value to stakeholders.

Introduction

Aims/objectives of this document

This document serves as the final report for the project AI and smart vision for house management. This document aims to provide the client stakeholder, as well as our industry mentor and any additional readers an overall view of the whole project. From client information, requirement gathering to the intricacies of the designs, product creation as well as problems faced during the project.

Key points to be addressed

This document will focus on firstly the Clients (who they are, what business problems they are facing, their current system). From that information, we will then provide our own assessment of their problems, what opportunities they have, and how our project can help them achieve this goal. We will then highlight their specific requirements, which we obtained through direct meetings and start the project. The project will go through a planning / researching stage, where we research and find the proper solutions that fit the requirements and budget constraint, as well as how to properly deploy the solution to match. Then we will go into deeper details on the development process of the project. And finally, we will reflect on the final product, as well as give future directions for the project.

Project overview/background

Client/company background description

NTT e-MOI JSC, a subsidiary of NTT East Japan, specializes in leveraging cutting-edge technology to drive business development and digital transformation for their 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 emphasizes 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)

Organisation’s current system(s) or situation

NTT e-MOI JSC has developed a property management solution that currently provides basic monitoring features. This system requires the business team to perform manual inspections and updates to track house changes. This approach, while functional, is becoming increasingly insufficient as the client base grows and the need for more efficient, accurate, and scalable solutions becomes apparent.

Manual inspections are time-consuming and prone to human error, leading to inefficiencies and inconsistencies in property monitoring. The lack of an automated system means that inspectors must physically visit each property to check for any changes or violations, resulting in delays and increased operational costs. These inefficiencies can have serious implications for property owners and regulatory bodies, as significant changes or potential violations might be missed or detected too late.

Problems, opportunities, objectives, expected benefits high-level scope statement

Problems

- **Short-Term**: The current manual inspection process is inefficient and prone to errors, leading to delays and increased operational costs. Inspectors face challenges in covering all properties comprehensively, resulting in missed detections and inconsistencies in reporting.
- **Long-Term**: The lack of an automated system limits scalability and the ability to respond quickly to changes, affecting customer satisfaction and regulatory compliance. The organization risks falling behind competitors who adopt advanced technologies for property management.

Opportunities

- **Short-Term**: Implementing an AI and Smart Vision system for house management can significantly reduce inspection time and increase accuracy (Monna et al., 2021). This will streamline operations and improve the reliability of property monitoring, providing immediate cost savings and operational efficiency.
- **Long-Term**: By adopting advanced technologies such as GPS data, satellite imagery, computer vision, and spatial data processing (Pan et al., 2020), NTT e-MOI can establish itself as a leader in innovative property management solutions. This will enhance the company's competitive edge, attract new clients, and open up opportunities for expanding their service offerings. Additionally, the automated system will enable better data-driven decision-making, leading to improved customer satisfaction and long-term partnerships.

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 on 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, 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.=

Scope

1.1 Analyzing Housing Images from Urban Areas

The app will implement advanced computer vision techniques to analyze housing images from urban environments, prioritizing 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.

1.2 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**: Utilizing 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 create 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.

1.3 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.

Solution Research and Development

Systems Architecture

Research

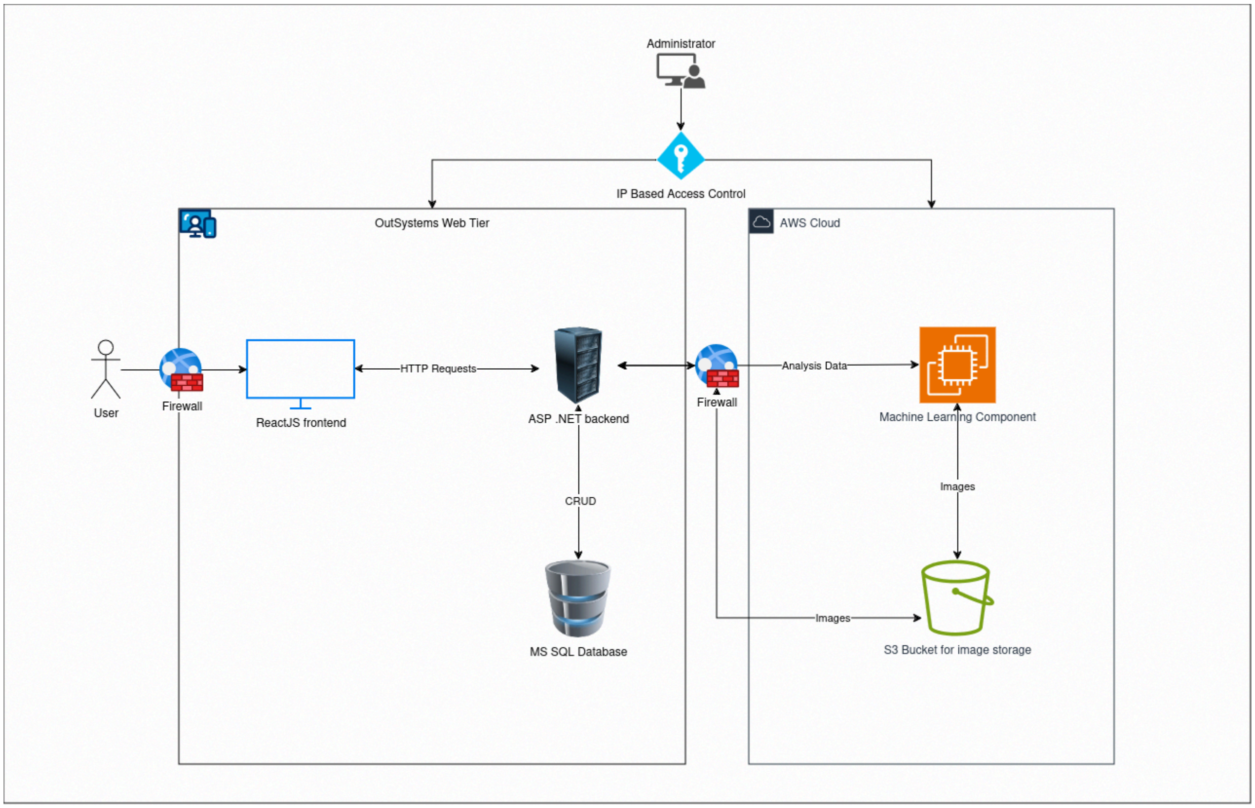
For the project's infrastructure architecture, there were several key points that we needed to address. Firstly, per the requirement, the solution is a web application that helps the user, in this case a property manager, manage properties through satellite images and analysis generated through a Machine Learning model. Secondly, the end user needs to be able to input data from previous work onto the system. And finally, the system needs to be secure, so that only people with proper privileges are able to access the application.

Our first consideration is a monolithic design, where everything is hosted on a singular server. This has the benefit of simplicity. With everything on a singular host, it is very simple to deploy, manage as there is less work to connect different parts of the applications together. However, this architecture has several issues. Firstly, this creates a single point of failure. If anything were to happen to the server, everything goes down, from the front end to back end as well as any data stored on the server, all will become inaccessible for the duration that the server is down for. Secondly, having only a single server also means that everything will share resources. This requires a lot of overhead planning when provisioning infrastructure, and it also comes at the cost of inflexibility, as increasing / decreasing resources will add additional overhead. After considerations, we have concluded that while the simplicity in deployment is very valuable, the risk associated with having everything sharing a single set of resources is too big, given the resource hogging nature of Machine Learning, which is a big selling point of our application.

Our second consideration is to move the Machine Learning component completely off of the web application component. This ensures that the Machine Learning part of the application can run freely, independently of the web component. This ensures that the Machine Learning side can use as many resources as needed without needing to consider the web application. Conversely, the web application is not affected by the Machine Learning department. This, while adding overhead as it means more servers to manage, also allows these components to be worked on / deployed / configured independently, making it highly suitable for both development and deployment of the application.

We went on with this architecture for the first phase of the development process. However, during this process, another problem arised. In the Machine Learning department, the images used for both the training process and the actual application are huge in size, and will only grow further in size as time goes on. While we can manually increase the hard drive size of the Machine Learning Server, this process requires a lot of manual labor, is very time consuming and error prone, sometimes the errors could be pretty fatal as it touches upon the server's configuration itself. So, we need to move these static images to a 3rd party storage system that is managed, can scale seamlessly and infinitely (practically speaking). We considered some Cloud Storage options like Azure Block Store, Google Cloud Storage, but we finally ended up on Amazon Simple Storage Service (S3). A time tested Cloud Storage system from Amazon that specializes in Block storage, is managed, infinitely scalable, secure and extremely high availability. To better integrate the Machine Learning component with the S3 storage system, we also decided to move the whole component to Amazon Elastic Compute Cloud (EC2). This leverages the inter-AWS network to allow for faster, more secure connection to the storage system, as well as a more centralized management interface for the infrastructure.

Design



Development

Web Application

For the web application, we have opted for a full stack kit provided by [OutSystems](). We went with this technology for a couple of reasons. One, familiarity. Our team is very experienced in working with OutSystems due to past projects, which will speed up the process of development and ensure familiarity. Secondly, OutSystems come packaged with their own Cloud Hosting service, which allows for instantaneous deployment, with fully managed infrastructure. That means less time spent configuring servers and dealing with dependencies, as they are all managed by the OutSystem cloud itself. This Cloud Hosting service also comes in a free tier, which our web application will not exceed due to the small number of projected users, and all data stored externally.

Machine Learning Component

For hosting the Machine Learning model for the application, we decided to use a t3.medium EC2 instance, which will give us 2 vCPUs and 4 GB of RAM, which will be just about enough for the machine learning model to work, while still maintaining within budget. We decided to use Ubuntu 24.04 LTS (Long term support) as the operating system due to its long term guaranteed support, numerous highly available documentation and integration options and of course, familiarity from our developers. The Machine Learning model will run through a MiniConda (Anaconda Documentation) virtual environment. This is a special python environment that ensures consistent python dependencies and behavior for our model to run on.

The Machine Learning component will also serve an interface for administrators to upload images for custom processing and communication with the web tier. This interface will also be served through MiniConda and leverages the FastAPI python framework.

Finally, we will set up security measures for the server. Firstly, we set up an ssh key for the server. Anyone who wants to access the server through SSH will need to provide this key in the SSH request in order to be authenticated. Secondly, we will lockdown SSH access even further by only allowing SSH requests from administrator's IP ranges. This configuration is achievable using AWS Security Groups (see: Control traffic to your AWS resources using security groups - Amazon Virtual Private Cloud). Thirdly, for web application access, we will open port 80 (HTTP) for web application access. It will also only be enabled to authorized IP ranges to lock down the application.

After all the configuration is complete and the Machine Learning component is up and running, we will snapshot the image so that we will be able to quickly rebuild the server if anything were to go wrong.

S3 Bucket

The final part of the infrastructure setup is to deploy an S3 bucket to store static images. We set up an S3 bucket on AWS, and lock down all public access to the bucket, this ensures that the bucket is completely private. Next, the bucket policy will be modified to allow access only from the Web Application tier and the Machine Learning model, through the use of an access key and a secret key, which acts as the authorization method for access to the bucket. This bucket is now ready, can scale infinitely and highly available.

Outcome

The outcome of the infrastructure building process is the architecture we mentioned in the design. An OutSystems full stack website that leverages analysis and data from a Machine Learning Component that runs independently of everything, and all static images will be stored in Amazon S3. All access will be restricted to only authorized individuals through an IP based filter system and SSH access is further monitored through SSH keypairs.

Web Application

Research

In the initial phase of developing the house detection website, the focus was on selecting a development platform that ensures flexibility, high performance, and stability. After evaluating various options, a platform was chosen that could handle the complex processing tasks required for analyzing satellite images while providing a robust and scalable foundation for future growth. Additionally, emphasis was placed on designing an interactive and engaging user interface (UI) and user experience (UX). The goal was to create an intuitive environment where end users could easily navigate and perform essential tasks, such as viewing historical data and tracking changes in house structures over time. Finally, priority was given to establishing a stable and efficient connection to the database. This connection is crucial for accurately retrieving house data and drawing bounding boxes around detected houses, ensuring users receive reliable and up-to-date information.

Design

The design of the house detection website centers around four main pages: the homepage, dashboard page, export page, and house mapping page. Each page serves a distinct purpose and offers specific functionalities to enhance the user experience. The homepage is designed to allow users to view various sections, with the capability to create or delete sections as needed. This flexibility ensures that users can manage their data efficiently and tailor the interface to their specific needs.

The dashboard page is dedicated to data visualization, providing users with comprehensive analytics for each section and across multiple sections. This feature enables users to gain insights into the changes in house structures over time, facilitating data-driven decision-making. The export page offers robust functionality for exporting data in multiple formats, including Excel, CSV, and JSON, as well as generating reports in PDF format. This ensures that users can easily share and utilize the data in their preferred formats for further analysis or reporting purposes.

The house mapping page is a key feature of the website, displaying the mappings of each house on an interactive map. Users can select different years for the same area to track changes in house structures over time, with the added ability to zoom in and out for a detailed view. Additionally, the page includes options to toggle between the background map and house bounding boxes, allowing users to customize their view and focus on specific details. This interactive design ensures that users can effectively monitor and analyze house changes with precision and ease.

Development

The development of the house detection website was carried out using OutSystems, a low-code platform that enables rapid application development with minimal hand-coding. OutSystems streamlines the development process by providing a visual development environment, pre-built templates, and drag-and-drop functionalities. This allows developers to quickly create, deploy, and manage applications while maintaining high standards of performance and scalability. The platform also integrates seamlessly with existing systems and databases, ensuring a smooth workflow and efficient data management.

Ensuring fully tested and secure connections was a top priority during the development phase. Rigorous testing protocols were implemented to guarantee the reliability and security of the connections between the application and the database. Additionally, loading indicators were applied where necessary to enhance user experience by providing visual feedback during data retrieval and processing. The development process also focused on achieving smooth file and data handling, ensuring that data transformations and file operations were performed efficiently and without errors. This meticulous approach to development ensures that the application operates seamlessly, providing users with a reliable and intuitive tool for tracking changes in house structures over time.

Outcome

The outcome of the development process is a fully functional and engaging user interface that meets the initial design goals. The interface is intuitive and interactive, allowing users to easily navigate through the various sections and perform essential tasks with minimal effort. The application successfully detects and draws bounding boxes around 90% of houses in every map, demonstrating a high level of accuracy and reliability in house detection from satellite images.

Additionally, the data and report export functionalities are smooth and unsusceptible to error. Users can effortlessly export data in multiple formats, such as Excel, CSV, and JSON, and generate comprehensive reports in PDF format. This feature ensures that the data can be easily shared and utilized for further analysis or reporting purposes. Furthermore, the connections and user data are safeguarded with high security measures, ensuring that sensitive information is protected from unauthorized access. The overall result is a robust and reliable application that effectively meets the needs of users for tracking house changes over time.

Machine Learning

Research

The initial phase of developing our AI-based house detection pipeline primarily involved evaluating viable methodologies. These methodologies can be broadly categorized into two types: object detection and semantic segmentation. Object detection provides robust house detection using bounding boxes, which also facilitates counting the number of houses. However, the diverse shapes of houses make bounding boxes suboptimal for accurately capturing house boundaries. Additionally, this method does not enable pixel-level precision in calculating house areas. Conversely, semantic segmentation offers pixel-level accuracy by classifying each pixel of an image as either belonging to a house or not. This precision allows for accurately delineating house shapes and calculating their areas.

Following this evaluation, we aimed to collect datasets for our model. We utilized two distinct datasets: the Inria Aerial Image Labeling Dataset [1] and the WHU Building Dataset [2]. Despite the varying image sizes in these datasets, we processed them by segmenting the images into 512x512 patches. This preprocessing yielded approximately 10,000 labeled images.

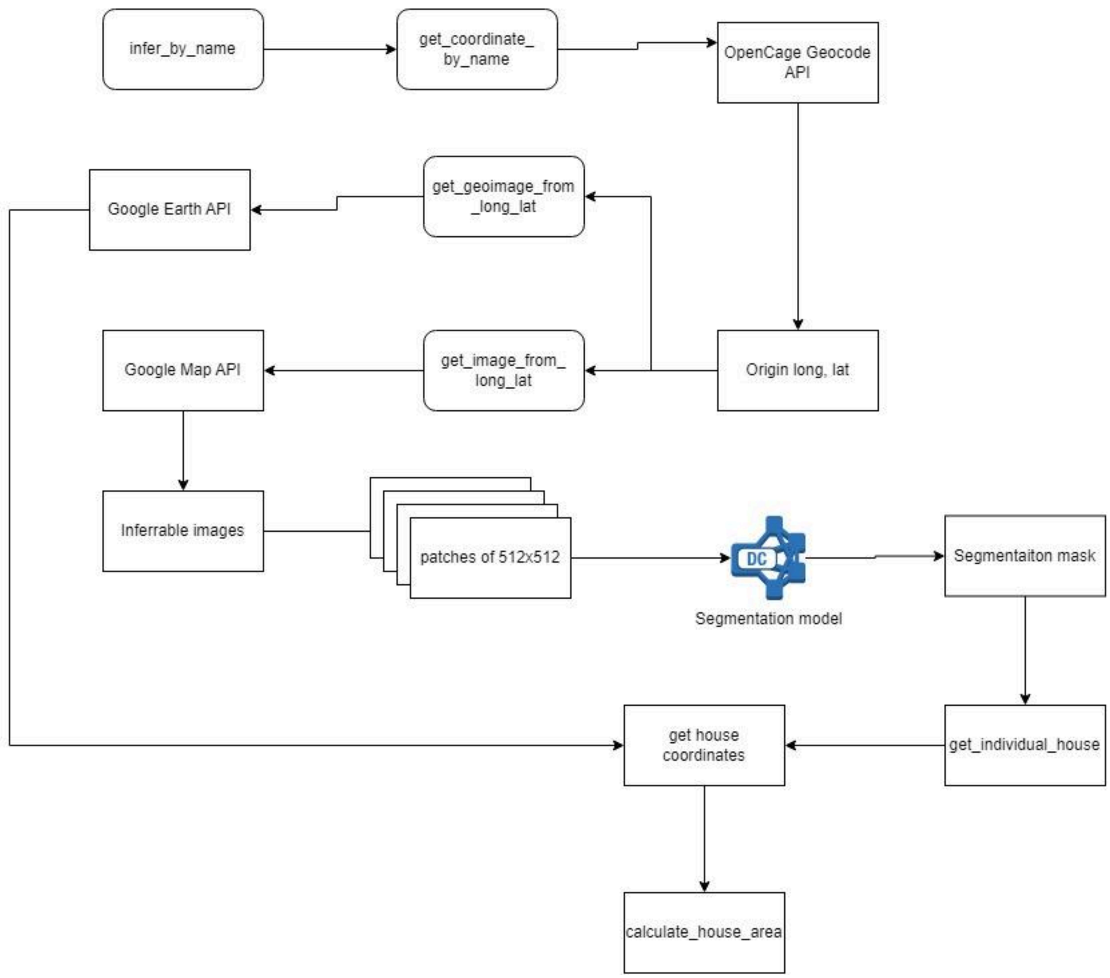
Subsequently, we proceeded with model selection, which involves choosing both an encoder and a decoder. The encoder functions as a feature extractor, producing a high-dimensional array that encapsulates the image information. The decoder then uses this feature vector to reconstruct an array where each pixel is classified as either 0 or 1, indicating whether the pixel belongs to a house (1) or the background (0).

In addition to model training, we need to address two primary challenges to achieve the objectives of our house management system:

1. Inferring House Properties, Such as Areas: To accurately infer house properties, particularly areas, we need to utilize the coordinates of house edges and apply the Haversine formula. This requires geo-metadata, which is available only in .TIF file format provided by satellite images. However, satellite images are either costly or lack the necessary quality for accurate inference. Although quality aerial images can be manually extracted using the Google Earth app, this method is labor-intensive and impractical for large-scale operations. Thus, finding a cost-free solution remains a significant challenge for the team.

2. Automating Image Acquisition for Inference: The goal of our information system is to automate processes and minimize reliance on end-user-provided images, which are often inconsistent and inaccurate. Therefore, we must develop a method to automate image acquisition while ensuring high-quality images suitable for inference and obtaining geo-metadata in .TIF format to calculate areas. This automation is crucial to streamline operations and enhance the efficiency of our house management system.

Design



Development

The development phase comprises two subphases: training and building the inference pipeline. During the training subphase, we established an experimental environment that facilitated the combination of various encoders with different decoders. We standardized the image size to 512x512, as this dimension is optimal for our resource constraints.
The setup details are as follows:

Resources	
CPU	i7-4970k
GPU	RTX2070 8GB VRAM
RAM	32GB DDR3
Image size	512x512

Batch size	2
Sample size	~10,000

For the encoders, we selected three model families: ResNet, EfficientNet, and Mix Vision Transformer. Each of these models offers unique advantages in terms of architecture, pre-trained datasets, and inference speed.

Regarding the decoders, we utilized three model families: UNet, FPN, and MANet. These models differ architecturally, with UNet and FPN employing convolutional layers, while MANet incorporates Transformer layers.

The results of our experiments are detailed below:

Encoder	Decoder	Average Val dataset IOU (higher is better)
ResNet	Unet	0.52
ResNet	FPN	0.61
ResNet	MANet	0.74
EfficientNet	Unet	0.66
EfficientNet	FPN	0.67
EfficientNet	MANet	0.77
Mix Vision Transformer	Unet	0.67
Mix Vision Transformer	FPN	0.73
Mix Vision Transformer	**MANet**	**0.81**

The second subphase, building the inference pipeline, presents a higher level of complexity. This phase addresses several critical tasks:

- Extracting high-quality images of sections for inference.
- Extracting .tif files of the same sections to obtain house coordinates, which are essential for calculating properties such as areas.
- Automating the extraction process to avoid end-user dependency.
- Ensuring the solution is cost-free.

To achieve these objectives, we rely on a set of APIs. Specifically, we use the OpenCage API [3] to generate coordinates for a section based on its name. Subsequently, we utilize the Google Maps API [4] to generate images of these coordinates. However, due to the Google Maps API's limitation of a maximum image size of 640x640 pixels, we developed a pipeline to stitch together image patches into a single, high-quality (4k) image for inference. For the geo-metadata file, we use the Google Earth Engine API [5], which, despite producing low-quality images, embeds metadata that can be aligned with the high-quality images from Google Maps to calculate coordinates. We then employ OpenCV to process the mask predictions, extracting individual houses and their contours/shapes. Finally, we calculate the areas using the Haversine formula.

Outcome

The outcome is a fully functional pipeline that employs the Mix Vision-Transformer and MANet architecture, hosted on a GPU-free EC2 instance. This setup achieves high accuracy and completes inference in approximately one minute from start to finish, given a section name. Remarkably, our solution incurs no cost. While there are limitations, such as inference speed, image quality, and the inability to automatically extract images from previous years, it remains a state-of-the-art solution within the given constraints.

Description, discussion and quality of recommendations

Systems Architecture

The Systems Architecture design, given the parameters, client requirements, budget constraint and time constraint, is very reasonable and practical. The decoupling of different components ensures little overlap, dependency injections and increases the application's availability. Moreover, this architecture also ensures independently setup and developed environments with no overlap, allowing smoother, faster development and deployment. The use of 3rd party block storage systems for images ensure the high availability, scalability and resilience of big image data that are also extremely accessible, easy to manage and highly secured.

However, there are still a few drawbacks with regards to this architecture model. Firstly, while different components are all separated, the underlying infrastructure only has 1 server. Given the small user load, 1 strong server is quite enough. However, if anything were to happen to these servers, the recovery methods are manually done by hand. This process, while sufficient enough given the use case and user requirements, is not very intuitive and easily error prone. Secondly, this architecture's scalability is quite vertical, as if the requirements were to change, and demand grows, growing server capacity could be a hussle, as scaling up will introduce downtime, while scaling out will introduce overhead on deployment and access. Both processes also need to be done manually.

However, the drawbacks are mainly concerns if the requirements were to drastically change in the near future. This architecture and its simplicity provides ease of management and development, as well as adhering strictly to the small budget given by the stakeholder, while still ensuring the promised application functionality.

Web Application

The recommendations for the house detection website are both appropriate and practical, tailored to address the specific needs and challenges identified during the research and design phases. The chosen development platform, OutSystems, is a low-code solution that offers flexibility, high performance, and stability, making it an ideal choice for rapid application development and efficient management of complex data processing tasks. This recommendation is well described, highlighting its benefits in terms of visual development, pre-built templates, and drag-and-drop functionalities, which streamline the development process and enhance scalability.

The advantages of using OutSystems include a faster development cycle, reduced need for extensive hand-coding, and seamless integration with existing systems and databases. This ensures that the development team can focus on optimizing the application's functionality and user experience. Additionally, the platform's robust security features help safeguard user data and maintain secure connections, addressing critical concerns related to data protection and privacy.

However, there are some disadvantages to consider. While OutSystems accelerates development and simplifies maintenance, it may impose certain limitations on customization and flexibility compared to fully custom-coded solutions. The reliance on a specific low-code platform may also introduce challenges related to vendor dependency and potential costs associated with licensing and support.

The extent to which the recommended solutions solve the identified problems and meet the requirements is significant. The user interface is designed to be fully functional and engaging, providing an intuitive environment for users to navigate and perform essential tasks efficiently. The high accuracy of house detection, with 90% of houses in every map being detected and bound, demonstrates the effectiveness of the chosen development approach and underlying algorithms.

Furthermore, the implementation of smooth and error-free data and report export functionalities ensures that users can easily share and analyze data, enhancing the overall utility of the application. The secure connections and robust data protection measures address critical security concerns, ensuring that user data is safeguarded against unauthorized access.

In summary, the recommendations provided for the development of the house detection website are well-described, appropriate, and practical. They address the identified requirements and challenges effectively, offering a balanced view of the advantages and potential disadvantages. The solutions proposed not only solve the immediate problems but also provide a scalable and secure foundation for future growth and enhancements.

Machine Learning

The AI component effectively addresses the identified problems by providing flexible methods for uploading images. By leveraging external free APIs and open-source deep learning models, we minimize development costs. Additionally, our solution offers end-to-end inference based solely on the name of the desired area. This triggers an automatic pipeline that extracts high-quality images, performs inference to detect individual houses, and uses mathematical models to accurately calculate house coordinates and areas.

While our AI pipeline does not yet support automated extraction of past-year images, which are only available through paid options, we still offer users the ability to upload their own historical images. These images can be manually extracted using the Google Earth Desktop app, allowing for some degree of flexibility.

Overall, the provided AI solution is both appropriate and practical, given the constraints set by the stakeholders. It effectively meets the requirements and addresses the challenges, offering acceptable trade-offs to enable a fully automated solution. Our system not only solves the primary issues of house management within a specific area but also provides a scalable framework that can be further enhanced and extended with additional features.

Future considerations

As the house detection website evolves, it will be essential to implement a structured change management process to handle updates and improvements effectively. This process should include thorough documentation, version control, and regular stakeholder communication to ensure that any changes are smoothly integrated without disrupting the existing functionality. Continuous feedback from users will be crucial in identifying areas for improvement and prioritizing new features or enhancements.

To further improve the development process, adopting Agile methodologies can be beneficial. Agile practices, such as iterative development, continuous integration, and regular sprint reviews, will help the development team respond to changing requirements quickly and maintain a high standard of quality. This approach will also facilitate better collaboration between developers, designers, and end users, ensuring that the application continues to meet user needs and expectations.

Future design enhancements should focus on expanding the application's functionality while maintaining an intuitive and user-friendly interface. Incorporating advanced data visualization techniques can provide deeper insights into house detection patterns and trends. Additionally, integrating machine learning algorithms could improve the accuracy of house detection and enable predictive analysis, offering users more sophisticated tools for tracking changes over time.

Exploring mobile-friendly designs and developing a companion mobile application can enhance accessibility, allowing users to monitor and analyze house data on the go. Ensuring that the interface is responsive and adaptable to various screen sizes will improve the overall user experience and broaden the application's reach.

The continued development and improvement of the house detection website will have several implications. First, as the application grows in complexity, maintaining robust security measures will be paramount. Regular security audits and updates will be necessary to protect user data and ensure compliance with relevant regulations.

Scalability will also be a critical consideration. As more users access the application and more data is processed, the underlying infrastructure must be capable of handling increased load without compromising performance. Cloud-based solutions and scalable architectures should be explored to accommodate future growth.

Further work will involve integrating additional data sources, such as different satellite imagery providers or other geospatial data, to enhance the comprehensiveness and accuracy of the house detection system. Collaboration with experts in geospatial analysis and machine learning can drive innovation and improve the system's capabilities.

In conclusion, while the current implementation of the house detection website meets the initial requirements and provides a solid foundation, ongoing improvements in change management, process optimization, design enhancements, and scalability will be essential. These considerations will ensure that the application remains relevant, reliable, and valuable to its users, adapting to their evolving needs and the dynamic nature of geospatial data analysis.

Conclusion

The AI and Smart Vision for House Management project was initiated to address the client's need for a more efficient, accurate, and scalable property management system. The primary goals were to develop a robust application capable of detecting house structures with high accuracy, facilitating efficient data handling and export, and improving overall user engagement and interaction.

The development of the house detection website has been a comprehensive process, covering research, design, development, and outcome evaluation. Utilizing OutSystems as the development platform provided a flexible, high-performance, and stable environment for creating a robust application. The design focused on user engagement and ease of navigation, resulting in an intuitive interface with well-defined functionalities across the homepage, dashboard, export, and house mapping pages.

The development phase ensured secure connections, efficient data handling, and smooth user interactions, culminating in a reliable and high-performing application. The outcome demonstrated a high level of accuracy in house detection and effective data export capabilities, meeting the initial goals and requirements.

The solution successfully addresses the business problem by significantly improving the efficiency, accuracy, and scalability of the property management process. The AI and vision technologies used have enhanced the detection accuracy and streamlined data management, thus replacing the previous manual processes with a more advanced and reliable system. The project's goals have been met through the development of a user-friendly, secure, and high-performing application that offers substantial value to its users.

Looking ahead, future considerations for change management, process improvement, and design enhancements will be crucial in maintaining and expanding the application's functionality. Implementing structured change management processes, adopting Agile methodologies, and incorporating advanced data visualization and machine learning techniques will drive continuous improvement. Continued focus on security, scalability, and user feedback will ensure that the application remains effective and relevant, adapting to evolving needs and technological advancements.

We would like to express our sincere gratitude to the client for their collaboration and support throughout the project. Your insights and feedback have been invaluable in shaping the final product. We look forward to continuing our partnership and driving further innovations that deliver long-term value to your stakeholders.

References

[1] Aerial Images dataset: https://project.inria.fr/aerialimagelabeling/
[2] WHU building dataset: http://gpcv.whu.edu.cn/data/building_dataset.html
[3] OpenCage Geocoder API: https://opencagedata.com/
[4] Google Maps API: https://mapsplatform.google.com/
[5] Google Earth Engine API: https://developers.google.com/earth-engine