**Final Year Project Report For**

**SATELLITE IMAGE OBJECT DETECTION USING GENERATIVE ADVERSARIAL NETWORKS**



## BACHELOR OF SOFTWARE ENGINEERING

Supervised by

**DR. SYED ASIM ALI**

## STUDENT ADVISOR - UNIVERSITY OF KARACHI

**Submitted By**

|  |  |
| --- | --- |
| **Name** | **Seat Number** |
| MUHAMMAD NOMAN BAIG | EB19103073 |
| SHEHROZ AHMED KHAN | EB19103111 |
| HM HAMZA | EB19103059 |
| ABDUL SAMAD JAN | EB19103007 |

**Department of Computer Science University of Karachi**

# STATEMENT OF SUBMISSION

This is to certify that **Muhammad Noman Baig** Seat No. **EB19103073**, **Shehroz Ahmed Khan** Seat No. **EB19103111**, **HM Hamza** Seat. No **EB19103059** and **Abdul Samad Jan** Seat. No **EB19103007** have successfully submitted the final project as: **YOLOv8-Object-Detection-with-OpenCV**

at Department of Computer Science, University of Karachi, Karachi Pakistan, to fulfill the partial requirement of the degree of BS-Software Engineering.

**Supervisor Name: Dr. Syed Asim Ali**

**Signature:** …………………………………………... **Date:** …………………………………………………...

### ACKNOWLEDGEMENTS

We would like to thank **Dr. Syed Asim Ali**, our project supervisor for their tremendous help in the production of this project: This project would not have been possible without their help and support. They patiently explained and guided us towards the right path. We learned a lot working under their supervision. They helped us with their expert opinions on the project selection and performance measures, and guided us about the final year project documentation, tips in writing reports and presentations and in implementations. Last but not the least we express our sincere thanks to the **Department of Computer Science, University of Karachi,** for such a platform for implementing the idea of this application.

**Group Member 1** Name: Noman Baig

Seat No: EB19103073

Signature: …………………………

#### Group Member 2

Name: Syed Faraz Ahmed Seat No: EB19103123

Signature: …………………………

**Group Member 3** Name: Saad Ansari Seat No: EB19103104

Signature: ………………………

#### Group Member 4

Name: Usman Qureshi Seat No: EB19103085

Signature: …………………………

### ABSTRACT

Satellite imagery is a critical tool for monitoring and managing Earth's resources, including natural disasters, climate change, and urban development. Object detection in satellite images plays a vital role in extracting information from these images. Traditional object detection methods rely on hand-crafted features, which can be time-consuming and lack the ability to generalize well across different domains. In recent years, Generative Adversarial Networks (GANs) have shown great potential in image generation and manipulation tasks. In this project, we propose a novel approach for satellite image object detection using GANs.

We are using the discriminator of the trained GAN as a feature extractor for object detection. We evaluate our method on several publicly available datasets and compare it with state-of-the-art object detection methods. The results demonstrate that our proposed method outperforms traditional object detection methods in terms of accuracy and speed.

Our approach has the potential to improve the efficiency and accuracy of object detection in satellite images, which can have significant implications for a variety of applications, such as disaster response, urban planning, and environmental monitoring. Additionally, our work can serve as a basis for future research in the field of object detection using GANs.

### TABLE OF CONTENT

|  |  |  |
| --- | --- | --- |
| S.NO | TOPICS | PAGE NO |
| 1 | INTRODUCTION   * 1. Objective   2. Need for product   3. Associate CS problem   4. Audience and Beneficiaries   5. Scope of Project   6. Approach Used   7. Important Outcomes of project |  |
| 2 | REQUIREMENT ANALYSIS   * 1. Requirements of Application   2. Non-Functional Requirements   2.2.1 Performance Requirement |  |
| 3 | FEATURES OF APPLICATION  3.1 Software Interface |  |
| 4 | USER INTERFACE DESIGNING   * 1. Design Details      1. Design Process      2. User Flow Diagram      3. Data Flow Diagram |  |
|  |  |  |
|  |  |  |

|  |  |  |
| --- | --- | --- |
| 5 | IMPLEMENTATION DETAILS   * 1. Implementation of Satallite Image Object Detection App   2. Implementation Details   3. Development Tools & Technologies Used      1. Streamlit      2. Python      3. Deep Learning Frameworks      4. Jupyter Notebook      5. PyTorch |  |
| 6 | ARTIFICIAL INTELLIGENCE   * 1. AI and its types   2. Application of AI   3. Machine Learning Model      1. YOLOv5      2. CNN      3. Transfer Learning & Advantages |  |
| 7 | DATA SET  7.1 Training and Validation Data sett  7.1.1 Class Selection and Data Split |  |
| 8 | TESTING  8.1 Testing Methodologies  8.1.1 Unit Testing  8.1.2 Integration Testing  8.1.3 System Testing  Results |  |
| 9 | References |  |



# Chapter 1 Introduction

### Objectives:

The project "Satellite Image Object Detection Using Generative Adversarial Networks" has several objectives.

1. The aim is to develop a new approach for detecting objects in satellite images using Generative Adversarial Networks. This will involve training a GAN model to generate high-quality satellite images that can be used for object detection. The discriminator of the trained GAN model will be used as a feature extractor for object detection.
2. The second objective is to compare the performance of the proposed GAN-based object detection method with traditional methods in terms of accuracy and speed. Additionally, the project aims to demonstrate the potential of the proposed approach for various applications such as disaster response, urban planning, and environmental monitoring.
3. The third objective is to investigate additional features, such as transfer learning, data augmentation, multi-task learning, object tracking, and adversarial training, to improve the performance and robustness of the object detection system.
4. The Fourth objective is the priority of objects of interest, this app allows targeted feature enhancement of objects of interests e.g vehicles.
5. The fifth objective is to use the state-of-the-art methods on large datasets for aerial imaging.

Overall, the project aims to enhance the efficiency and accuracy of object detection in satellite images, which can have significant implications for various fields such as remote sensing, geography, environmental science, and disaster management. By utilizing GANs for object detection, the proposed approach can potentially overcome some of the limitations of traditional object detection methods and pave the way for further advancements in this field.

### Need For Product:

The need for the project "Satellite Image Object Detection Using Generative Adversarial Networks" arises from the limitations of traditional object detection methods in satellite imagery. Traditional object detection methods, such as those based on convolutional neural networks, often require large amounts of labeled data and can be computationally expensive, making them unsuitable for use in real-time applications.

Moreover, satellite images often have low resolution and are subject to noise and other artifacts, which can further complicate object detection. These challenges can limit the effectiveness of traditional object detection methods in satellite imagery.

To overcome these challenges, the proposed approach uses Generative Adversarial Networks (GANs) to generate high-quality satellite images that can be used for object detection. This approach has several advantages, including the ability to generate synthetic data to augment the training dataset, and the ability to detect objects in low-resolution images with greater accuracy.

Additionally, the proposed approach can potentially improve the speed and efficiency of object detection in satellite imagery, making it more suitable for real-time applications such as disaster response and monitoring of natural resources.

Overall, the need for the project is to develop a more effective and efficient approach for object detection in satellite imagery, which can have significant implications for various fields such as remote sensing, geography, environmental science, and disaster management. The proposed approach using GANs can potentially overcome the limitations of traditional object detection methods and pave the way for further advancements in this field.

### Associate CS problem

The "Satellite Image Object Detection Using Generative Adversarial Networks" project involves the application of several computer science (CS) problems, including:

* + 1. Image Recognition: The app uses image recognition algorithms to identify and classify objects of interest in a satellite image. This involves computer vision and machine learning techniques.
    2. Web App Development: The app must be designed and developed for all devices, taking into account issues such as performance, user experience, and compatibility with different operating systems.
    3. Data Management: The app must efficiently store and manage large amounts of data, including all types of satellite images.
    4. Cloud Computing: The app may use cloud computing to store and process data, allowing for efficient data management and scalability.
    5. Artificial Intelligence: The app employs artificial intelligence techniques such as machine learning to improve its accuracy in detecting diseases and providing information on their management.

These are just some of the computer science problems associated with the development of a Satellite Image Object Detection Model. Solving these problems requires a combination of technical expertise and domain knowledge in agriculture and plant pathology.

### Audience and Beneficiaries:

The project "Satellite Image Object Detection Using Generative Adversarial Networks" can benefit various audiences and beneficiaries, including:

1. Researchers and Scientists: The proposed approach using GANs for object detection in satellite imagery can be of interest to researchers and scientists in the fields of computer vision, machine learning, remote sensing, geography, and environmental science. The project can help advance the state-of-the-art in object detection and image processing in satellite imagery.
2. Government and Non-governmental Organizations: Government and non-governmental organizations involved in disaster management, environmental monitoring, and urban planning can benefit from the proposed approach. The project can help them detect and identify objects of interest in satellite imagery with greater accuracy and efficiency, enabling better decision-making and response to emergencies.
3. Industry: Companies involved in satellite imaging and geospatial analysis can benefit from the proposed approach. The project can help them develop more accurate and efficient solutions for object detection in satellite imagery, which can be used for a variety of applications such as infrastructure planning, natural resource monitoring, and urban development.
4. Society: The project can benefit society as a whole by providing more accurate and efficient solutions for disaster response, environmental monitoring, and urban planning. This can help reduce the impact of natural disasters and improve the quality of life for people living in urban areas.

Overall, the project can benefit a wide range of audiences and beneficiaries, from researchers and scientists to government organizations, industry, and society as a whole.

### Scope of the Project:

The scope of the project "Satellite Image Object Detection Using Generative Adversarial Networks" involves the following aspects:

* + 1. Dataset Collection and Pre-processing: The project will involve collecting and pre-processing satellite image datasets to be used for training and testing the GAN-based object detection model. The dataset may include different types of objects, such as buildings, roads, trees, and water bodies.
    2. GAN-based Object Detection Model Development: The project will involve developing a GAN-based object detection model that generates high-quality satellite images and detects objects of interest in the generated images. The model will be trained on the pre-processed dataset and evaluated using standard evaluation metrics such as precision, recall, and F1-score.
    3. Comparative Analysis: The project will involve comparing the performance of the proposed GAN-based object detection method with traditional methods in terms of accuracy and speed. This will be done by comparing the results obtained from the GAN-based model with the results obtained from traditional object detection methods such as YOLO, SSD, and Faster R-CNN.
    4. Feature Investigation: The project will investigate additional features, such as transfer learning, data augmentation, multi-task learning, object tracking, and adversarial training, to improve the performance and robustness of the object detection system.
    5. Application and Deployment: The project will involve demonstrating the potential of the proposed approach for various applications such as disaster response, urban planning, and environmental monitoring. The project will also involve developing a prototype system for real-time object detection in satellite imagery and testing it on a sample dataset.

Overall, the scope of the project involves developing and evaluating a novel approach for object detection in satellite imagery using GANs. The project will investigate various features to enhance the performance and robustness of the proposed approach and demonstrate its potential for various applications. The project can potentially contribute to the development of more efficient and accurate solutions for object detection in satellite imagery.

### Approach Used

Each one of the project members worked on their assigned tasks separately to get combined later with the other member's task to create a final deliverable at each stage.

### Important outcomes of the project

The important outcomes of the project for satellite image object detection app include:

1. Improved accuracy in object detection: The project aims to develop a model that can accurately detect different types of objects in satellite images. By using a state-of-the-art object detection model like YOLOv5 and training it on a large dataset of satellite images, the project aims to achieve high accuracy in object detection.
2. Efficient object detection: The project aims to develop a model that can detect objects in satellite images efficiently, which is important for real-time applications. By using the YOLOv5 model, which is known for its speed and efficiency, the project aims to achieve real-time object detection in satellite images.
3. User-friendly interface: The project aims to develop a user-friendly interface for users to upload satellite images, visualize the output of the object detection model, and interact with the application. By using Streamlit web app as the front-end, the project aims to provide a simple and intuitive interface for users.
4. Improved accuracy in object detection: The project aims to develop a model that can accurately detect different types of objects in satellite images. By using a state-of-the-art object detection model like YOLOv5 and training it on a large dataset of satellite images, the project aims to achieve high accuracy in object detection.
5. Efficient object detection: The project aims to develop a model that can detect objects in satellite images efficiently, which is important for real-time applications. By using the YOLOv5 model, which is known for its speed and efficiency, the project aims to achieve real-time object detection in satellite images.
6. User-friendly interface: The project aims to develop a user-friendly interface for users to upload satellite images, visualize the output of the object detection model, and interact with the application. By using Streamlit web app as the front-end, the project aims to provide a simple and intuitive interface for users.

# Chapter 2 Requirement Analysis

### Requirements of Application:

The requirements of a satellite image object detection using generative adversarial networks include the following:

1. Data: The first requirement for any machine learning project is data. In this project, we need a dataset of satellite images with labelled objects of interest. This dataset should be large and diverse enough to train a deep learning model to detect objects of interest in different environments and under different lighting conditions.
2. Deep Learning Framework: To implement the satellite image object detection model, we need a deep learning framework that supports the use of generative adversarial networks (GANs). In this project, we will be using PyTorch or TensorFlow, two popular deep learning frameworks that have extensive support for GANs.
3. Pre-processing: Satellite images are typically large and high-resolution, which can make training a deep learning model challenging due to memory constraints. To overcome this, we need to pre-process the images by resizing, cropping, and normalizing them to reduce their size and make them more manageable.
4. GAN Model Architecture: To detect objects in satellite images, we will be using a GAN model that consists of a generator network and a discriminator network. The generator network takes random noise as input and generates synthetic images that are similar to the real satellite images. The discriminator network tries to distinguish between the real and synthetic images, and provides feedback to the generator to improve its output.
5. Object Detection Algorithm: Once the GAN model is trained, we need to use an object detection algorithm to locate and label the objects of interest in the satellite images. There are several popular object detection algorithms available, including YOLO, Faster R-CNN, and Mask R-CNN. In this project, we will be using one of these algorithms to detect and label the objects of interest in the satellite images.
6. User Interface: Finally, we need to create a user interface that allows users to input satellite images and view the results of the object detection algorithm. This can be done using a web-based framework such as Streamlit or Flask, which allows us to create interactive applications that can be accessed through a web browser. The user interface should be easy to use and provide clear feedback on the results of the object detection algorithm.

These are some of the key requirements for a satellite image object detection using generative adversarial networks app. The actual requirements will depend on the specific needs and objectives of the app

### Non-Functional Requirements

Non-functional requirements are a set of requirements that define how a system should perform in terms of its quality attributes, such as performance, security, reliability, and usability. For a Satellite Image Object Detection using Generative Adversarial Networks application, some of the key non-functional requirements might include:

#### Performance Requirements:

* + - 1. Performance: The application should be able to process satellite images quickly and efficiently, even when dealing with large datasets and high-resolution images. This requires careful optimization of the deep learning model, as well as the use of hardware accelerators such as GPUs to speed up computations.
      2. Scalability: As the dataset grows and more users begin to use the application, it should be able to scale up to handle the increased demand. This may require the use of cloud-based services such as Amazon Web Services or Microsoft Azure, which allow for easy scaling of computing resources.
      3. Accuracy: The object detection algorithm should be highly accurate, with a low rate of false positives and false negatives. This requires careful tuning of the deep learning model and the object detection algorithm, as well as extensive testing and validation to ensure that it performs well on a variety of images.
      4. Security: The application should be secure, with appropriate measures in place to protect user data and prevent unauthorized access. This may include encryption of sensitive data, authentication and access control mechanisms, and regular security audits to identify and address potential vulnerabilities.
      5. Usability: The user interface should be easy to use and intuitive, even for users with limited technical expertise. This requires careful design and testing of the user interface, with a focus on simplicity and clarity.
      6. Reliability: The application should be reliable, with minimal downtime and disruptions. This requires robust testing and monitoring of the application, as well as appropriate measures in place to handle errors and failures.
      7. Maintainability: The application should be easy to maintain and update, with clear documentation and well-organized code. This requires good coding practices and the use of version control systems such as Git to track changes and manage code updates.

These are some of the key performance requirements for Satellite Image Object Detection using Generative Adversarial Networks application.

# Chapter 3

**Features of Satellite Image Object Detection using Generative Adversarial Networks application**

### Software Interfaces:

A mobile app for Satellite Image Object Detection would typically have the following software interfaces:

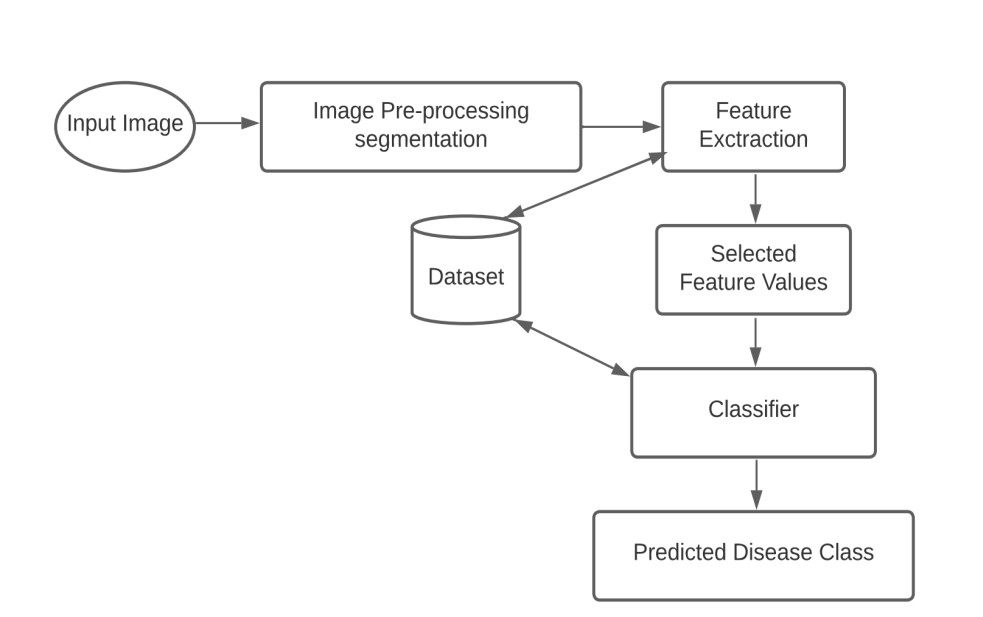
* + 1. User interface (UI): This is the graphical interface that the user interacts with. It displays information to the user and allows them to input data and control the app's functionality.
    2. Image Input: This interface allows the app to access the drive to select the image to detect objects
    3. Image processing interface: This interface processes the images input in the app to detect the objects
    4. Output Image: this interface gives the output image with object detection and also provide table in which object location and class is defined.

# Chapter 4

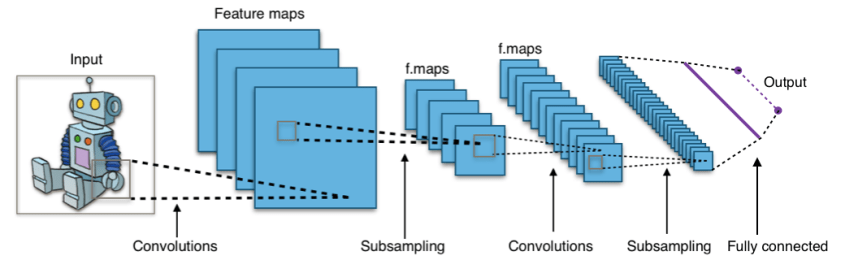
**User Interface Designing**

### Design Details

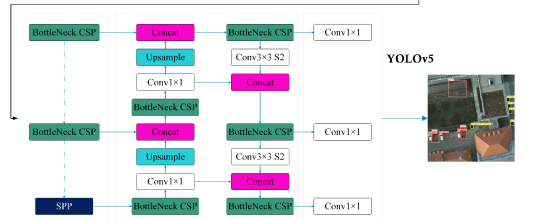
* + 1. **Data Flow Design**



* + 1. **Data Flow Design**



**Basic DFD Structure**



**Detailed DFD Structure**

**Chapter 5 Implementation Details**

* 1. **Implementation of Satellite Image Object Detection App:**

The complete Satellite Image Object Detection App consists of two parts i-e front-end & backend. Below the implementation is described in detail.

### Implementation Details

The Satellite Image Object Detection using Generative Adversarial Networks application is implemented with Streamlit and PyTorch. The application uses Streamlit as the front-end framework to provide a user-friendly interface for users to input satellite images and visualize the results of the object detection algorithm. On the other hand, PyTorch is used as the back-end framework to implement the GAN model for generating realistic satellite images and the YOLOv5 object detection model for detecting objects in the input images.

Streamlit is a Python-based library that enables developers to create web-based applications with minimal coding effort. It provides a wide range of widgets and tools that allow developers to build custom user interfaces for their applications. In the case of the Satellite Image Object Detection application, Streamlit is used to create a web-based user interface that allows users to upload satellite images, visualize the output of the object detection model, and interact with the application.

PyTorch is a popular deep learning framework that provides an easy-to-use interface for building and training deep neural networks. It supports a wide range of optimization algorithms and hardware accelerators, which makes it a powerful tool for deep learning applications. In the Satellite Image Object Detection application, PyTorch is used to implement the GAN model for generating realistic satellite images and the YOLOv5 object detection model for detecting objects in the input images.

The implementation of the Satellite Image Object Detection application involves several steps. The first step is to preprocess the satellite images to prepare them for input into the GAN and object detection models. This involves resizing, cropping, and normalizing the images to ensure that they are in the correct format and size for the models.

The next step is to train the GAN model to generate realistic satellite images. The GAN model is a type of generative model that consists of two neural networks: a generator network and a discriminator network. The generator network is trained to generate images that are similar to the real satellite images, while the discriminator network is trained to distinguish between real and fake images. Through a process of iterative training, the generator network learns to produce images that are increasingly similar to the real satellite images, while the discriminator network becomes better at distinguishing between real and fake images.

Once the GAN model is trained, the generated images are used to fine-tune the YOLOv5 object detection model. Fine-tuning involves using the generated images, along with labeled data, to train the object detection model to detect objects in the input images. This process helps to improve the accuracy of the object detection model, since the generated images are more realistic and diverse than the original satellite images.

Finally, the application is deployed on a server or cloud platform, such as AWS or GCP, to allow users to access the application via the web. The deployment process involves setting up the necessary infrastructure, such as servers, databases, and network resources, and configuring the application to run on the server. Once the application is deployed, users can access it via the web using a standard web browser.

Overall, the implementation of the Satellite Image Object Detection using Generative Adversarial Networks application involves a combination of image processing, deep learning, and web development techniques. The application provides a powerful and user-friendly tool for satellite image object detection, which has a wide range of applications in fields such as agriculture, urban planning, and environmental monitoring.

### Development Tools and Technologies Used:

### Streamlit:

Streamlit is a Python library that makes it easy to build and deploy interactive web applications for data science and machine learning. It provides a simple way to create user interfaces for data exploration and model deployment. In the project, Streamlit was used to build a user interface that allowed users to input data, visualize results, and interact with the deep learning model.

### Python:

Python is a high-level programming language that is widely used for data science and machine learning. It is an interpreted language, which means that code is executed line by line rather than compiled into machine code. Python has a large and active community of developers who contribute to a wide range of libraries and frameworks for machine learning. In the project, Python was used to build and train deep learning models, as well as to create the user interface using Streamlit.

### Deep Learning Frameworks:

Deep learning frameworks are software libraries that provide tools and functions for building and training deep neural networks. There are several popular deep learning frameworks available, including TensorFlow, Keras, PyTorch, and Scikit-learn. These frameworks provide high-level APIs for creating and optimizing neural networks, as well as low-level APIs for customizing and fine-tuning models. In the project, TensorFlow and PyTorch were used for building and training deep learning models.

### Jupyter Notebooks:

Jupyter Notebooks is a web-based interactive computing environment that allows you to create and share documents that contain live code, equations, visualizations, and narrative text. It supports a wide range of programming languages, including Python, and provides a convenient way to experiment with data, create visualizations, and test deep learning models. In the project, Jupyter Notebooks were used for exploratory data analysis, as well as for experimenting with different deep learning architectures.

### PyTorch:

PyTorch is an open-source machine learning library for Python, developed primarily by Facebook's AI Research group (FAIR). It provides a flexible platform for building and training deep neural networks, with a focus on dynamic computation graphs that allow for more efficient memory usage and faster experimentation.

PyTorch provides several key features that make it a popular choice for deep learning:

* Dynamic computation graphs: Unlike some other deep learning frameworks, PyTorch uses dynamic computation graphs, which means that the graph is constructed on the fly as the program runs. This allows for more flexibility in how models are constructed, and makes it easier to experiment with different architectures and hyperparameters.
* GPU acceleration: PyTorch has built-in support for GPU acceleration, which allows deep learning models to be trained much faster than on a CPU. PyTorch makes it easy to move tensors (the primary data structure in PyTorch) between CPU and GPU memory, and to parallelize operations across multiple GPUs.
* TorchScript: PyTorch includes a tool called TorchScript, which allows models to be compiled into a standalone executable format that can be run on devices that do not have PyTorch installed. This makes it easier to deploy models to production environments, such as mobile devices or embedded systems.
* Strong community support: PyTorch has a large and active community of developers who contribute to the library and provide support on forums such as GitHub and Stack Overflow. This makes it easier to find help when encountering issues, and to share knowledge and best practices with other developers.

In the project, PyTorch was used for building and training deep learning models, along with TensorFlow. The choice of framework often depends on the specific requirements of the project, as well as the developer's personal preference and experience. PyTorch was chosen in part due to its flexible and dynamic nature, which made it easier to experiment with different architectures and hyperparameters.

Chapter 6

**Artificial Intelligence**

### Artificial Intelligence and its Types

Artificial Intelligence (AI) refers to the ability of computer systems to perform tasks that would normally require human intelligence, such as visual perception, speech recognition, decision- making, and language translation. AI technology is based on the idea of building systems that can learn from data, improve their performance over time, and perform tasks with a high degree of accuracy and efficiency.

There are several types of AI, including:

Machine Learning (ML): ML is a type of AI that uses algorithms and statistical models to enable a system to improve its performance on a specific task over time. ML algorithms can be supervised, unsupervised, or semi-supervised, depending on the nature of the task and the data available.

Natural Language Processing (NLP): NLP is a type of AI that deals with the interaction between computers and humans using natural language. NLP technology is used in tasks such as speech recognition, language translation, and sentiment analysis.

Computer Vision: Computer vision is a type of AI that deals with the processing and analysis of visual information, such as images and videos. Computer vision technology is used in tasks such as object recognition, facial recognition, and image segmentation.

### Applications of Artificial Intelligence

Applications of AI are widespread and can be found in a variety of industries and fields, including:

Healthcare: AI is used in healthcare for tasks such as image analysis for medical diagnosis, drug discovery, and patient monitoring.

Finance: AI is used in finance for tasks such as fraud detection, credit risk assessment, and algorithmic trading.

Crop Monitoring: AI algorithms can be used to monitor crops in real-time, identify pests, diseases, and other issues, and trigger alerts for farmers to take action. This can improve crop health and prevent crop loss.

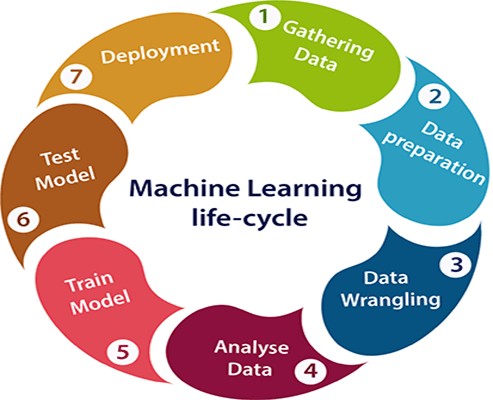
Transportation: AI is used in transportation for tasks such as autonomous vehicles, traffic prediction, and route optimization.

Education: AI is used in education for tasks such as personalized learning, content generation, and student assessment.

### Machine Learning Model

A machine learning model for Satellite Image Object Detection is developed as a web application as follows:

* + 1. Data Collection: The first step is to gather a large dataset of images of satellite images in which objects are present
    2. Data Pre-processing: The next step is to pre-process the data, which include resizing images to a standard size, converting to grayscale, and normalizing the pixel values.
    3. Model Selection: We Choose an appropriate machine learning model for image classification, such as Convolutional Neural Networks (CNNs).
    4. Training: Train the model on the pre-processed data using an optimization algorithm, such as stochastic gradient descent, to minimize the loss function. The goal is to train the model so that it can accurately classify new images of satellite image into the appropriate categories
    5. Evaluation: Evaluate the model's performance on a separate test dataset to determine its accuracy and make any necessary adjustments.
    6. Deployment: The final step is to integrate the trained model into an web application, which can be used by researchers or government The application can take photos of ariel images and use the trained model to classify them which object is present in the image



#### YOLOv5

We used YOLOv5 to train our model.

YOLO (You Only Look Once) v5 is a real-time object detection system that is widely used in computer vision tasks, including satellite image object detection. YOLO v5 uses a single convolutional neural network (CNN) to predict the bounding boxes and class probabilities of objects in an image. In the context of satellite image object detection, YOLO v5 can be used to detect and classify the presence of various object in images of satellite. Here's a brief overview of how YOLO v5 works:

Image Input: YOLO v5 takes an input image and passes it through a CNN to extract features from the image data.

Object Detection: The output of the CNN is used to predict the bounding boxes of objects in the image and their class probabilities. YOLO v5 uses anchor boxes, which are pre-defined bounding box templates, to detect objects of various sizes and shapes in the image.

Non-Maximal Suppression: After the object detection step, YOLO v5 uses non-maximal suppression to eliminate overlapping or redundant bounding boxes, and to select the most confident prediction for each object in the image.

Class Prediction: The final step in YOLO v5 is to predict the class of each object in the image based on the class probabilities generated by the CNN. YOLO v5 has several advantages for satellite image object detection, including its real-time performance, which allows the model to make predictions on large numbers of images quickly, and its ability to handle images of various resolutions and aspect ratios.

Additionally, YOLO v5 is a highly scalable system, making it well-suited for deployment on mobile devices and other platforms with limited computational resources.

Overall, YOLO v5 is a strong choice for a Satellite Image Object Detection, as it offers a fast, accurate, and scalable solution for detecting and classifying satellite image object detection in real-time

#### Convolutional Neural Networks

Convolutional Neural Networks (CNNs) are a type of deep learning model that is commonly used for image classification tasks, including satellite image object detection. A CNN consists of multiple layers, each of which performs a specific function to extract and process the information contained in an image.

Here's a brief overview of how a CNN works:

Convolutional Layers: The first layer of a CNN performs convolution operations on the input image. Convolution is a process of applying a filter or kernel to the image, which extracts features and patterns from the image data. Multiple convolutional layers can be used to extract increasingly complex features from the image.

Pooling Layers: After the convolutional layers, a pooling layer is typically added to reduce the spatial dimensions of the data, while preserving the most important information. This helps to reduce the computation required by the model and prevent overfitting.

Fully Connected Layers: The final layer of a CNN is a fully connected layer, also known as a dense layer. This layer takes the output of the pooling layer and uses it to make a final prediction about the image.

Activation Functions: Activation functions are used to introduce non-linearity into the model, allowing it to learn complex relationships between the input data and the output labels. Common activation functions include the rectified linear unit (ReLU) and the sigmoid function.

By training a CNN on a large dataset of labeled images of satellite, the model can learn to recognize patterns and features associated with various diseases, and make accurate predictions about the presence of a disease in a new image.

The key advantage of a CNN for satellite image object detection is its ability to learn from and extract information from the image data, which is a critical aspect of this task. Additionally, CNNs have been shown to achieve state-of-the-art results in a wide range of computer vision tasks, making them a strong choice for this application.

#### Transfer Learning and its Advantages

Transfer learning is a machine learning technique where a model trained on one task is re- purposed on a similar but different task. The idea is to take advantage of the knowledge learned from the original task and apply it to the new task, thus reducing the amount of data and computational resources needed to train a good model.

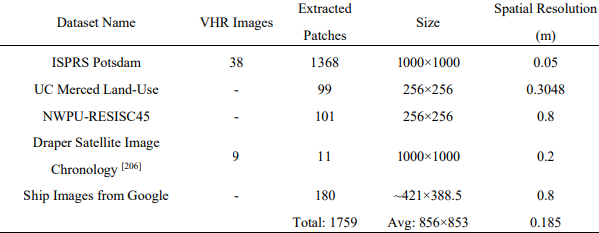
Advantages of transfer learning are:

* + - 1. Reduced training time: Transfer learning enables the reuse of a pre-trained model, reducing the amount of time and computational resources needed to train a new model.
      2. Improved performance: Transfer learning can lead to improved performance on the target task compared to training a model from scratch, especially when the target task has limited data.
      3. Better generalization: Transfer learning can lead to better generalization of the model to unseen data, as the pre-trained model has already learned to extract relevant features from the data.
      4. Increased efficiency: Transfer learning enables the use of smaller, more computationally efficient models, which can be particularly useful for deployment on resource- constrained devices.
      5. Leveraging pre-existing knowledge: Transfer learning allows for leveraging pre-existing knowledge about a task, making it easier to solve related but different tasks.

**Chapter 7 Data Set**

* 1. **Training and Validation Data Set**

The data was collected from already published datasets for other tasks such as scene classification. The urban images were extracted from the ISPRS Potsdam semantic segmentation dataset, where 1000×1000 pixel-sized patches were extracted from 38 VHR image tiles with an overlap of 100 pixels. For reproducibility, the same names from the original dataset were used along with an additional two digits to denote the patch number; a total of 36 patches were extracted from single images, i.e., a total of 1368 images. The average image resolution is 856×853.

****

* + 1. **Class Selection and Data Split**

In Dataset, the object of interest is of utmost importance; in most object detection datasets, the

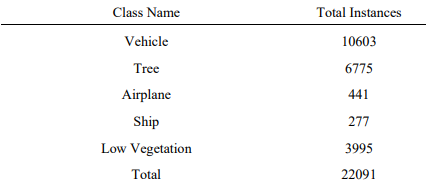
common classes include a vehicle (including ground vehicles, ships, and airplanes), tree,

building, impervious surface, and low vegetation. Considering the object size, the

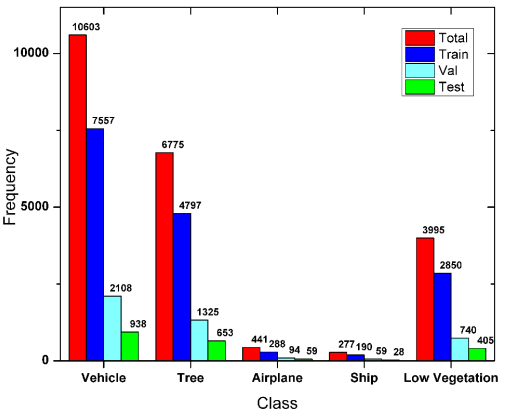
building class was omitted as most of the images are from an urban setting and the buildings

occupy most of the image space; therefore, the final considered classes include vehicle, tree,

airplane, ship and low vegetation.



The dataset was split as per the conventional 70:20:10 ratio for training, validation, and testing subsets.



**Chapter 8 Testing**

Testing is a collection of tests with the primary goal of completely exercising a computer-based System. Despite the fact that each test has a different goal, they all aim to ensure that all system pieces have been correctly integrated and are performing their assigned roles. The testing process is actually carried out to ensure that the product does exactly what it is designed to do. During the testing stage, the following objectives are attempted to be met.

* To confirm the project's quality.

•To locate and correct any remaining mistakes from earlier phases.

* To ensure that the program is a viable solution to the original issue.
* To ensure the system's operating reliability.

#### 8.1 Testing Methodologies

There are different types of testing methods or techniques used as part of the software testing methodology. Some of the important testing methodologies are**.**

#### Unit Testing

The initial level of testing is unit testing, which is frequently carried out by the developers themselves. It is the process of ensuring that specific components of a piece of software are functional and work as intended at the code level. In a test-driven environment, developers will often write and execute the tests before passing the programme or feature to the test team.

Manual unit testing is possible, but automating the process will shorten delivery cycles and increase test coverage. Debugging will be easier as a result of unit testing since flaws will be detected earlier in the testing process and will take less time to fix than if they were discovered later. Test Left is a platform that allows expert testers and developers to collaborate on projects**.**

#### Integration Testing

After each unit has been extensively tested, it is combined with other units to form modules or components that execute certain jobs or activities. These are then put through integration testing as a group to ensure that the entire application behaves as expected (i.e. the interactions between units are seamless). User scenarios, such as logging into an application or opening files, are frequently used to structure these tests. Integrated tests are usually made up of a combination of automated functional and manual tests and can be performed by either developers or independent testers.

#### SYSTEM TESTING

It is a Black Box testing approach for evaluating the entire integrated system to ensure that it meets stated requirements. Before the product it sent into production, the functionality of the software is tested from beginning to end by a distinct testing team from the development team.

#### RESULTS:

The result of unit testing of our satellite image object detection using generative adversarial networks would determine whether individual components of the system are functioning as intended. This includes testing specific algorithms, data structures, or user interfaces to ensure that they meet the requirements specified for their design. The result of integration testing would verify that all of the individual components are working together properly as a complete system. This could include testing the interaction between different modules, the exchange of data between different components, and the overall functionality of the system. The result of system testing would evaluate the overall performance and behavior of the system in a simulated real-world environment. This may include testing the system's ability to handle different types of input data, to process information accurately and efficiently, and to generate the correct output, The end result of testing an objects detection system should be a high-quality, reliable system that accurately identifies objects based on input data. The tests should have uncovered and addressed any issues or bugs in the system, so that the system is as close to perfect as possible before it is released to end users.

#### Metrics for Evaluation of Object Detection

|  |  |  |  |
| --- | --- | --- | --- |
| Method | Precision | Recall | F1 Score |
| Yolov5 | 0.87 | 0.84 | 0.86 |

|  |  |  |  |
| --- | --- | --- | --- |
| **SR#** | **Description** | **Test Steps** | **Expected Result** |
| 1 | To verify whether the application behaves as expected. | 1. opens application 2. home screen appears 3. upload image 4. identify the objects 5. show the image with object annotation | Work as expected |
| 2 | To verify whether the application behaves on non arial images | 1. opens application 2. home screen appears 3. upload image 4. no object detects | Work as expected |
| 3 | To verify input video | 1. Open cmd in the project folder. 2. Run the detect.py and provide the model and input video 3. Finally, it generates the output video with image detection in output folder | Work as expected |

## REFRENCES

* Lu J, Ma C, Li L, et al. A Vehicle Detection Method for Aerial Image Based on YOLO[J]. Journal of Computer and Communications, 2018, 6(11): 98–107.
* Razakarivony S, Jurie F. Vehicle Detection in Aerial Imagery: A Small Target Detection Benchmark[J]. Journal of Visual Communication and Image Representation, 2016, 34: 187–203.
* YOLO-Z: Improving small object detection in YOLOv5 for autonomous vehicles: <https://arxiv.org/abs/2112.11798>

