SAVITRIBAI PHULE PUNE UNIVERSITY,PUNE

A

PROJECT STAGE-II REPORT ON

“People (person) and Trees detection from the drone’s real-world data”

SUBMITTED TOWARDS THE

PARTIAL FULFILLMENT OF THE AWARD OF THE DEGREE OF

BACHELOR OF TECHNOLOGY (COMPUTER ENGINEERING)

BY

MR. Tejas Kshirsagar	**[UCS20M1074]**
MR. Ninad Nimbhorkar	**[UCS20M1093]**
MR. Pratik Patil	**[UCS20M1101]**
MR. Om Sabne	**[UCS20M1113]**

UNDER THE GUIDANCE OF

Dr. A. V. BRAHMANE

Department of Computer Engineering

SANJIVANI COLLEGE OF ENGINEERING

KOPARGAON-423603

(AN AUTONOMOUS INSTITUTE)

2023-2024

[10/2023-2024]

Sanjivani College of Engineering, Kopargaon-423603

(An Autonomous Institute)

DEPARTMENT OF COMPUTER ENGINEERING

CERTIFICATE

This is to certify that the project entitled

'People (person) and Trees detection from the drone's real- world data"

Submitted by

MR. Tejas Kshirsagar	**[UCS20M1074]**
MR. Ninad Nimbhorkar	**[UCS19M1093]**
MR. Pratik Patil	**[UCS20M1101]**
MR. Om Sabne	**[UCS19M1113]**

is a bonafide work carried out by students under the supervision of Dr. A. V. BRAHMANE and it is submitted towards the partial fulfillment of the requirement of Bachelor of Technology (Computer Engineering).

During the Academic Year 2023-24

Dr. A. V. BRAHMANE
[Internal Guide]

Dr. S. R. Deshmukh
[Project Co-ordinator]

Dr. D. B. Kshirsagar
[Head of Dept.]

Dr. A. G. Thakur
[Director]

Signature of Internal Examiner

Signature of External Examiner

PROJECT APPROVAL SHEET

A

PROJECT STAGE- II REPORT

ON

"People (person) and Trees detection from the drone's real- world data"

Is Successfully Completed By

MR. Tejas Kshirsagar	**[UCS20M1074]**
MR. Ninad Nimbhorkar	**[UCS20M1093]**
MR. Pratik Patil	**[UCS20M1101]**
MR. Om Sabne	**[UCS20M1113]**

At

DEPARTMENT OF COMPUTER ENGINEERING

SANJIVANI COLLEGE OF ENGINEERING,KOPARGAON – 423603

(AN AUTONOMOUS INSTITUTE)

SAVITRIBAI PHULE PUNE UNIVERSITY, PUNE

ACADEMIC YEAR 2023-24

Dr. A. V. BRAHMANE	**Dr. S. R. Deshmukh**
[Internal Guide]	[Project Co-ordinator]
Dr. D. B. Kshirsagar	**Dr. A. G. Thakur**
[Head of Dept.]	[Director]

ACKNOWLEDGEMENT

"People (person) and Trees detection from the drone's real-world data" has been a subject with tremendous scope to research, which leads one's mind to explore new heights in the field of Computer Engineering, and its miscellaneous applications. We dedicate all our project work to our esteemed guide **Prof. A. V. Brahmane**
whose interest and guidance helped us to complete the work successfully as well as he has provided facilities to explore the subject with more enthusiasm. This experience will always encourage us to do our work perfectly and professionally. We also extend our gratitude to **Dr. D. B. Kshirsagar (H.O.D. Computer Department)**
. We express our immense pleasure and thankfulness to all the teachers and staff of the Department of Computer Engineering, Sanjivani College of Engineering, Kopargaon for their cooperation and support. Last but not least, we thank all others, and especially our parents and friends, who in one way or another, helped us in the successful completion of this project. We extend special gratitude to **Dr. S. R. Deshmukh** for their constant support and guidance throughout the project. Additionally, we acknowledge the invaluable contribution of **Dr. A. G. Thakur** and the **college management** for providing us with the platform and infrastructure necessary for our endeavors."

MR. TEJAS KSHIRSAGAR
MR. NINAD NIMBHORKAR
MR. PRATIK PATIL
MR. OM SABNE

(B.TECH. COMPUTER)

ABSTRACT

This study describes the creation of a cutting-edge computer vision system that can recognize and differentiate between individuals (people) and trees in real-world data collected by drones. This system harvests priceless insights from aerial imagery using cutting-edge machine learning algorithms and image analysis techniques, opening up new prospects for applications in forestry management, disaster response, and ecological study. In our project, specialized machine learning models are developed that are intended for accurate object recognition and classification. These models are painstakingly designed to recognize and distinguish between human forms and different tree species, even in difficult outdoor settings with variable lighting and dynamic viewpoints frequently found in drone captured video. It enables automated tree detection and classification in forestry management, making resource planning and monitoring more efficient and effective. In instances involving disaster response, it helps in the quick identification of people, potentially saving lives and speeding up rescue efforts. This system can be a valuable resource for ecologists conducting research on tree populations and human-environment interactions. By enabling drone-based data analysis and advancing cutting-edge technologies, our research helps to improve resource allocation and decision-making across a variety of situations. Our study addresses significant real-world difficulties and provides a way towards more effective and efficient data-driven solutions by improving the identification of people and trees within various environments.

Contents

List of Figures

List of Tables

Chapter 1

INTRODUCTION

1.1 Problem Definition

Unmanned aerial vehicles (UAVs), whether automated drone's or remotely piloted, play vital roles in aviation, free from on-board crews. They excel in endurance, maintaining controlled flight for extended periods. Equipped with diverse sensors, drones gather data encompassing speed, distance, temperature, wind, chemicals, light, sound, and magnetic fields. The collected data holds immense value for drone professionals and organizations. In fields like agriculture, construction, infrastructure, and maintenance, UAVs empower businesses to capture real-time, high-quality data, enhancing operational efficiency. This data is transmitted via secure, high-bandwidth links. In cases of interrupted transmission due to distance or environmental constraints, satellite communication expedites the process. Crucially, drones excel in detecting people and trees, facilitating surveillance of roads, construction sites, and more, offering insights and improving safety. This technology showcases the wide-reaching impact of UAVs in modern applications.

1.2 Literature Review/Relevant Theory

Literature survey is the most important step in software development process. Before developing the tool, it is necessary to determine the time factor, economy and company strength. Once these things are satisfied, ten next steps are to determine which operating system and language can be used for developing the tool. Once the programmers start building the tool the programmers need lot of external support. This support can be obtained from senior programmers, from book or from websites. Before building the system, the above consideration is taken into account for developing the proposed system.

1. Conference/Journal: 2020 International Conference on Artificial Intelligence, Big Data, Computing and Data Communication Systems (icABCD)

Paper Title: IEEE Xplore Paper - "Deep Learning-Based Object Detection for UAV Imagery"

Author: Katleho L Masita; Ali N Hasan; Thokozani Shongwe

Object detection continues to play a significant part in computer vision theory, study and practical application. Conventional object detection algorithms were primarily derived from machine learning. This involved the design of features for describing the object's characteristics followed by an integration with classifiers. In recent years, the application of deep learning (DL), and more specifically Convolutional Neural Networks (CNN) have elicited a great advancement and promising progress, and has therefore, received much attention on the global stage of research about computer vision. [1] This paper conducts a review about some of the most important and recent developments and contributions that have been made towards research in the use of deep learning in object detection. Moreover, as evidently demonstrated, the findings of numerous studies suggest that the application of deep learning in object detection much surpasses conventional approaches focused on handcrafted and learned features.

2. Conference/Journal: 2020 International Conference on Artificial Intelligence, Big Data, Computing and Data Communication Systems (icABCD)

Paper Title: Tree Species Classification of Drone Hyperspectral and RGB Imagery with Deep Learning Convolutional Neural Networks

Author: Somayeh Nezami, Ehasas khoramshashi

This study investigates the classification of three major tree species—pine, spruce, and birch—using 3D convolutional neural networks (3D-CNN) in a boreal forest environment. The study assesses both the individual and combined performance of 3D-CNN models trained with hyperspectral (HS), Red-Green-Blue (RGB), and canopy height model (CHM) data.[2] The 3D-CNN model that incorporates RGB and HS layers gets the maximum classification accuracy, according to the data, which is noteworthy [1]. The study displays impressive classification metrics, with the top 3D-CNN classifier achieving producer accuracy rates for pine, spruce, and birch trees, respectively, of 99.6.

1.3 Scope

1. **Search and Rescue Operations:**
   Drones equipped with people detection algorithms can be used in search and rescue missions, particularly in remote or disaster-stricken areas. They can identify individuals in need of assistance, improving the efficiency and effectiveness of rescue efforts.

2. **Environmental Monitoring and Conservation:**
   Drones are valuable tools for monitoring tree populations and forest health. Tree detection algorithms can assess deforestation, track tree growth, and identify tree species. This data aids in forest management and conservation efforts.

3. **Urban Planning and Infrastructure Development:**
   Drones can assist in urban planning by detecting trees and people in urban areas. This information can be used to plan green spaces, assess the impact of construction projects on the environment, and optimize city layouts for improved quality of life.

4. **Wildlife Conservation:**
   In the context of wildlife conservation, drones can be used to monitor the movement of both people (such as poachers) and animals. This helps protect endangered species by detecting and preventing illegal activities.

1.4 Objectives

- Search and Rescue: The primary objective is to precisely locate and identify individuals in distress during search and rescue missions, particularly in remote or disaster-stricken regions. This objective is pivotal for saving lives and expediting emergency response efforts.

- Environmental Monitoring: Objectives encompass evaluating tree populations, closely monitoring forest health, and tracking long-term changes in vegetation. These aims facilitate effective forest management, bolster conservation endeavors, and enable early intervention against diseases or deforestation.

- Agriculture and Precision Farming: The foremost goal is to continuously monitor crop health, swiftly identify pest infestations or diseases, and fine-tune agricultural practices. Additionally, detecting people aids in efficiently managing labor resources in extensive farming operations.

- Wildlife Conservation: Objectives entail vigilant monitoring of wildlife habitats, meticulous tracking of animal migrations, and early detection of human intrusions into protected areas. These aims are crucial for safeguarding endangered species and maintaining ecological equilibrium.

Chapter 2

REQUIREMENT ANALYSIS

Requirements Analysis or requirement engineering is a process of determining user expectations for new software or providing updates for previous products. These core points must be measurable, relevant and detailed. In the software engineering field this term is also called functional specifications. Requirements analysis mainly deals with communication with users or customers to determine system feature expectations, requirements and reduce convicts as demanded by various software users. Energy should be directed towards ensuring that the system or product conforms to user needs rather than attempting to turn user expectations to the requirements.

2.1 Requirement Specifications

Requirement specification describes the function and performance of the computer based system and constraints which govern its development. It can be a written document, a set of graphical models, a collection of scenarios, or any combination of above. These are of 3 types:

1. NR: Normal Requirements
2. ER: Expected Requirements
3. XR: Exciting Requirements

2.1.1 Normal Requirements

These are the requirements which are clearly stated by the customer so all these requirements will be present in the project for user satisfaction.

- NR1: System should analysis the people and trees in drone images.
- NR2: Appropriate output record of detection in the database..
- NR3: Detection of result of classes should accurate and precise.

2.1.2 Expected Requirements

These requirements are expected by the customer but not clearly stated by the customer. These are implicit types of requirements.

- ER1: System should be fast and reliable.
- ER2: System should have a neat and clean user interface.
- ER3: System should provide instruction of usage to the user.

2.1.3 Exciting Requirements

These requirements are not stated by the customer but externally provided by the developer in order to maintain a good relationship with the customer.

- XR1: Develop a real-time detection system that can process and analyze corn leaf images on the fly, enabling immediate response to disease outbreaks.

2.2 Validation of Requirements

Requirements validation is the process of checking that requirements defined for development, define the system that the customer really wants. To check issues related to requirements, we perform requirements validation. We usually use requirements validation to check error at the initial phase of development as the error may increase excessive rework when detected later in the development process.

In the requirements validation process, we perform a different type of test to check the requirements mentioned in the Software Requirements Specification (SRS), these checks include:

- Completeness checks
- Consistency checks

- Validity checks
- Realism checks
- Ambiguity checks
- Verifiability

The output of requirements validation is the list of problems and agreed on actions of detected problems. The lists of problems indicate the problem detected during the process of requirement validation. The list of agreed action states the corrective action that should be taken to fix the detected problem.

The development of software begins once the requirements document is ready. One of the objectives of this document is to check whether the delivered software system is acceptable. For this, it is necessary to ensure that the requirements specification contains no errors and that it specifies the users requirements correctly. Also, errors present in the SRS will adversely affect the cost if they are detected later in the development process or when the software is delivered to the user. Hence, it is necessary to detect errors in the requirements before the design and development of the software begins. To check all the issues related to requirements, requirements validation is performed. In the validation phase, the work products produced as a consequence of requirements engineering are examined for consistency, omissions, and ambiguity. The basic objective is to ensure that the SRS reflects the actual requirements accurately and clearly. The requirement checklist as follows:

1. Are all requirements consistent?
2. Are the requirements really necessary?
3. Is each requirement testable?
4. Does the requirement model properly reflect the information function and behaviour of the system to be built?

2.3 Functional Requirement

- System should take the input from the user.
- System should accept parameters like image etc.

- System should detect the required objects.
- System should classify the objects.
- System should give proper information.

2.4 Non-Functional Requirement

- System shall provide guidelines for user.
- System shall live for 24 hours.
- System shall backup the data.

2.5 System Requirements

Requirements specify what features a product should include and how those features should work. They help to define the test criteria, which is vital for verification and validation.

2.5.1 Hardware Requirements

1. RAM: 4GB (min)
2. Hard Disk :20 GB
3. Processor: Intel core i3 8th Gen
4. 64-bit CPU

2.5.2 Software Requirements

- Operating system: Windows 7,10,11.
- Browser: Chrome, Firefox
- IDE: Visual Studio Code or Jupiter
- Language: Python 3.7

Chapter 3

SYSTEM MODEL

Software process model is an intellectual demonstration of a process. It presents an explanation of a process. Process models may contain activities that are part of the software process. All software process models work on the five generic framework activities such as communication, planning, modelling, construction, and deployment. Each and every activity has its own functionality. The goal of the process model is to provide guidance for systematically coordinating and monitoring the tasks that must be accomplished in order to achieve the end product and the project objective.

3.1 Process Model

Software process model is an abstract representation of a process. The goal of process model is to provide guidance for systematically coordinating and controlling the tasks that must be performed in order to achieve the end product and the project objective. Incremental model is used as the process model in our system.

3.1.1 Incremental Model

Incremental model in software engineering is a one which associates the elements of waterfall model in an iterative manner. It delivers a series of releases called increments which provide progressively more functionality for the user as each increment is delivered. In the incremental model of software engineering, the waterfall model is frequently applied in each increment. The incremental model applies linear sequences in a required pattern as calendar time passes. Each linear sequence produces an increment in the work.

A, B, C are modules of Software Product that are incrementally developedand delivered. Every subsequent release of a module adds function to the previous release. This process is continued until the complete system is achieved

Phases of Incremental Model

1. Requirement analysis: In the very first phase of the incremental model, the product analysis expertise identifies the requirements. And the system functional requirements are understood by the requirement analysis team. To develop the software under the incremental model, this phase performs a crucial role.
2. Design Development: In this phase of the development, the design of the system functionality and the development method are finished with success. When software develops new practicality, the incremental model uses style and development phase.
3. Testing: In the incremental model, the testing phase checks the performance of each and every existing function as well as additional functionality. In the testing phase, the various methods are used to test the behaviour of each task.
4. Deployment: Once the product is tested, it is deployed in the production environment or first User Acceptance Testing (UAT) is done depending on the customer expectation. In the case of UAT, a replica of the production environment is created and the customer along with the developers does the testing.
5. Maintenance: After the deployment of a product on the production environment, maintenance of the product i.e., if any issue comes up and needs to be fixed or any enhancement is to be done is taken care by the developers.

3.1.2 Why to use Incremental Model

We used Incremental model for our system design because of reasons mention below:

- Major requirements must be defined: however, some detail can evolve with time.
- There is a need to get a product to market early.
- The software will be produced quickly during the software life cycle.
- It is flexible and less luxurious to change requirements and scope.
- Though the development stages changes can be done.

- This model is less costly than others.
- A customer can answer back to each building.

3.1.3 Advantages

1. Initial product delivery is faster.
2. Lower initial delivery cost.
3. Core product is developed leading i.e., main functionality is added in the first increment.
4. After each iteration, regression testing should be conducted. During this testing, faulty elements of the software can be quickly recognized because few changes are made within any single iteration.
5. It is generally easier to test and debug than other methods of software development because comparatively smaller changes are made during each iteration. This allows for more targeted and difficult testing of each element withinthe overall product.
6. With each release, a new article is added to the product.
7. Customers can reply to features and review the product.
8. Risk of changing requirements is reduced.
9. Workload is less.

3.1.4 Disadvantages

1. It requires good planning and designing.
2. Problems might be caused due to system architecture as such not all requirements are collected up front for the entire software lifecycle.
3. Each iteration phase is rigid and does not overlap each other.

Rectifying a problem in one unit requires correction in all the units and consumes a lot of time. It may arise affecting to system architecture because not all necessities are gathered up front for the entire software life cycle.

3.2 System Breakdown Structure

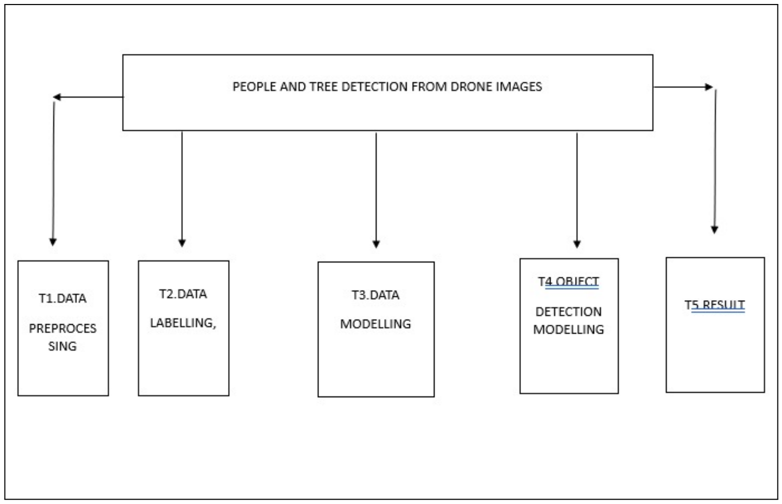

Figure 3.1: System Breakdown Structure

3.3 Module Details

These are the following modules which will be executed in this system.

1. Users
2. System

3.3.1 User

- User can give the input for particular parameter.
- User can view the predicted win or loss probability and score.

1. Parameter Input:

   Users input parameters for people and tree detection in drone images, specifying person size thresholds, confidence levels, color filters, tree height ranges, foliage density, and spatial distribution preferences. General parameters include image quality, noise reduction, and acceptable viewing angles, tailoring the system to specific detection needs.

The model for object detection using deep neural networks (DNN) is a process that involves the following steps:

2. Data preprocessing: This step involves preparing the input images for the DNN model. This may include resizing, cropping, augmenting, normalizing, and encoding the images. The goal is to make the images suitable for the DNN model and improve its performance. Data preprocessing may also involve labeling the images with the ground truth bounding boxes and class labels for the objects of interest. This can be done manually or using some annotation tools. The labels are usually stored in a JSON file along with the image paths.

3. DNN model training:

   This step involves training the DNN model on the preprocessed images and labels. The DNN model consists of a backbone network and a detection head. The backbone network is a convolutional neural network (CNN) that extracts features from the images. The detection head is a module that predicts the bounding boxes and class labels for the objects in the images. There are different types of detection heads, such as anchor-based, anchor-free, and transformer-based. The DNN model is trained using a loss function that measures the difference between the predicted and ground truth bounding boxes and labels. The loss function may also include some regularization terms to prevent overfitting. The DNN model is optimized using a gradient-based algorithm, such as stochastic gradient descent (SGD) or Adam. The DNN model training may also involve some training strategies, such as learning rate scheduling, data balancing, and model pruning.

4. Object detection:

   This step involves using the trained DNN model to detect objects in new images. The DNN model takes an image as input and outputs a set of bounding boxes and class labels for the objects in the image. The bounding boxes are usually represented by four coordinates (x, y, width, height) or (x1, y1, x2, y2). The class labels are usually represented by integers or strings.The DNN model may also output a confidence score for each bounding box, indicating the probability of the bounding box containing an object of the corresponding class. The object detection step may also involve some post-processing techniques, such as non-maximum suppression (NMS), which removes redundant or overlapping bounding boxes, and thresholding, which filters out low-confidence bounding boxes.

3.3.2 Admin Module

- Admin module will store details in the database.
- Admin module will produce a message.
- Admin module will send a message to the user.

1. Data Collection:

   Data collection is the process of gathering and measuring information from countless different sources.

2. Pre-processing:

   Preprocessing involves adding the missing values, the correct set of data, and extracting the functionality. Data set form is important to the process of analysis.

3. Model Building:

   (a) Data Splitting: Data splitting is when data is divided into two or more subsets. Typically, with a two-part split, one part is used to evaluate or test the data and the other to train the model. Data splitting is an important aspect of data science, particularly for creating models based on data.

   (b) Training Prediction Model: Training data is the data you use to train an algorithm or machine learning model to predict the outcome you design your model to predict.

   (c) Testing Prediction Model: Test data is used to measure the performance, such as accuracy or efficiency,of the algorithm you are using to train the machine.

4. Performance analysis of machine learning algorithm:

   The performance of any ML algorithm may improve with the utilization of a distinct set of features in the same training dataset.

3.4 Project Estimation

In this section various calculation and estimations related to project has been calculated. The figure shows the system modules. The number of lines required for implementation of various modules can be estimated as follows

Sr. No	Modules	Estimated KLOC
1	Data Pre processing	0.6 KLOC
2	Data Labeling	0.6 KLOC
3	Data Modelling	0.8 KLOC
4	Object detection Module	0.3 KLOC
5	Result	0.4 KLOC
Total		2.7 KLOC

Figure 3.2: System Breakdown Structure

3.4.1 Estimation in KLOC

Estimation is the process of finding an estimate, or approximation, which is a value that can be used for some purpose even if input data may be incomplete, uncertain, or unstable. Estimation determines how much money, effort, resources, and time it will take to build a Specific system or product. The number of lines required for implementation of various modules can be estimated as follows.

Table 3.1: Estimation of KLOC

Sr.no	Task	Estimated KLOC
1	Data Collection And Preprocessing	0.6
2	Training Object model	0.5
3	Parameter Tunning	0.6
4	Testing Against Inputs	0.4
5	UI Building	0.6
6	Classification of Trees and People	0.6
	Total	3.3

3.4.2 Efforts

The Efforts required in person/month for implementation can be estimated as follows:

$$Effort = a * (LOC)^b$$

Where, $a = 3.2$ and $b = 1.05$.

Hence, we get:

$$Effort = 3.2 * (KLOC)^{1.05}$$

$$\Rightarrow Effort = 3.2 * (3.3)^{1.05}$$

$$\Rightarrow Effort = 11.2 \quad persons/month$$

3.4.3 Development time in months

We know that:

$$Time\ required = \frac{Efforts}{No.\ of\ Developers}$$

$$\Rightarrow Time\ required = \frac{11.2}{4}$$

$$\Rightarrow Time\ required = 2.8 \quad months$$

3.4.4 Total Time Required for Project Development

The total time required can be calculated as follows:

Table 3.2: Time Required for Project Development

Task	**Time Required**
Requirement Analysis and Design	2 months
Implementation and Testing	2.5 months
Total	4.5 months

Hence, total time required is nearly 4.5 months.

3.4.5 Number of Developers (N)

The project is assigned to a group of 4 people (developers). Hence, the number of developers is taken 4. The developers are as follows:

- D1: Tejas Kshirsagar
- D2: Ninad Nimbhorkar
- D3: Pratik Patil
- D4: Om Sabne

Chapter 4

SYSTEM ANALYSIS

4.1 Project Scheduling and Tracking

Project Preparation and Tracing is important because in order to build a complex system, many software engineering tasks occur in parallel, and the result of work performed during one task may have a reflective effect on work to be conducted in another task. The inter dependencies are very difficult to understand without a detailed schedule.

4.1.1 Project Work and Breakdown Structure (Analysis)

The project work is decomposed into the following work breakdown structure as a part of the analysis phase.

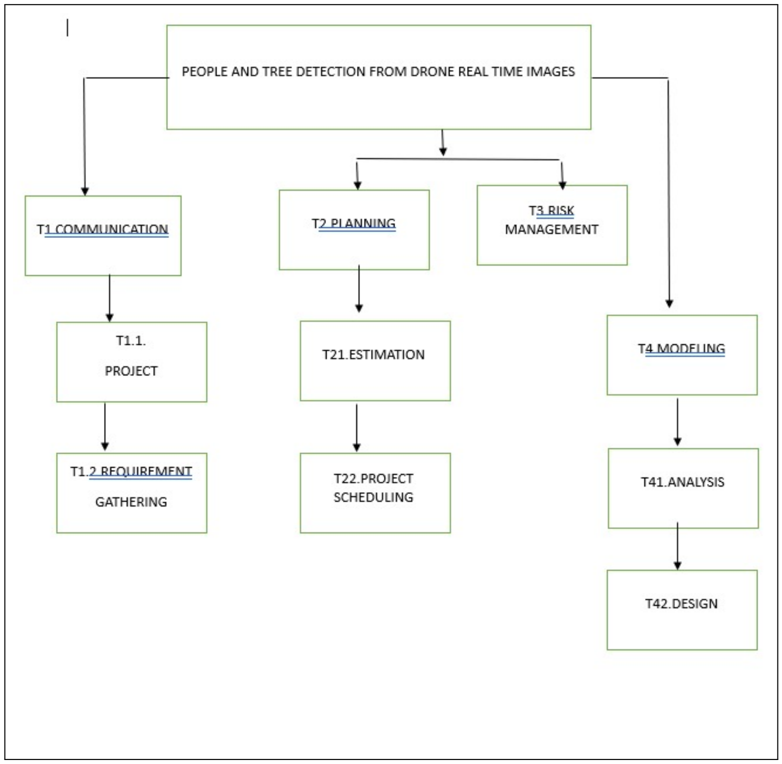

Figure 4.1: System Breakdown Structure

- **T1 : Communication** Software development process starts with the communication between customer and developer. According to need of project, we gathered the requirements related to project. Requirement gathering is an important aspect as the developer will come to know what customer expects from the project and also he can help a customer to know more features that can be added to project as he is a technical person. The most important thing needed is that communication should be smooth and clear that means developer should easily understand the demands of customer.
- **T2 : Planning** It includes complete estimation and scheduling (complete time line chart for project development). Before starting the project tasks should be scheduled that means there should be starting and ending date assigned for each and every task and developer should work harder to complete the required task within time chosen at the time of scheduling.
- **T3 : Modeling** It includes detailed requirements analysis and project design (algorithm, flowchart etc). Flowchart shows complete pictorial view of the project and algorithm is step by step solution of problem. Both flowchart and algorithm will be

helpful in knowing the overall view of project and serve as a base for development of whole project.

- **T4 : Risk Management** It is a process of identifying, organizing, assessing and controlling threats to some organizations' capitals and earnings which assets overall or partial software product or performance. These threats, or risk, could stem from a wide variety of sources, including financial uncertainty, legal liabilities, strategies, management errors, accidents and natural disasters.

4.2 Project work breakdown structure (Implementation)

Implementation is the stage of the project when the theoretical design is turned out into a working system.

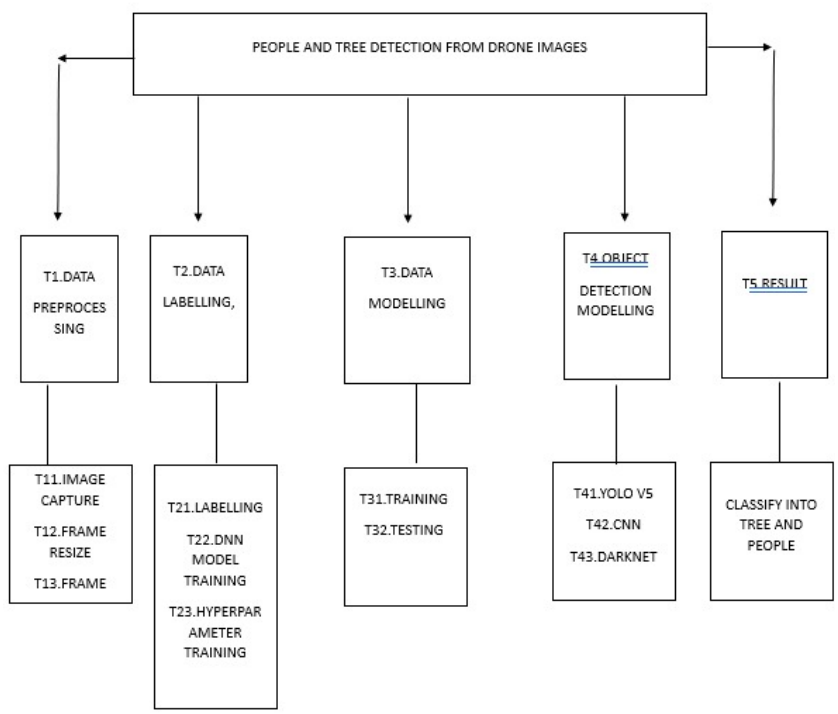

Figure 4.2: Breakdown Structure (Implementation)

Implementation is the stage of the project when the theoretical design is turned out into a working system. Thus, it can be considered to be the most critical stage in achieving a successful new system and in giving the user, confidence that the new system will work and be effective. The modules included in the system are:

User

- Users can easily give input parameters.
- Users can view the classification of detected people and trees.

1. Parameter Input: Users input parameters for people and tree detection in drone images, specifying person size thresholds, confidence levels, color filters, tree height ranges, foliage density, and spatial distribution preferences. General parameters include image quality, noise reduction, and acceptable viewing angles, tailoring the system to specific detection needs.
2. People Detection Outputs: - Detected Individuals: Locations and count of identified people. - Bounding Boxes: Visual markers outlining recognized individuals. - Confidence Scores: Confidence levels associated with each detection.
3. Trees Detection Outputs: - Detected Trees: Locations and count of identified trees. - Bounding Boxes: Visual markers outlining recognized trees. - Height Information: Estimated height of detected trees.
4. General Outputs: - Annotated Images: Visual feedback with annotations highlighting detected people and trees. - Confidence Reports: Detailed reports on confidence levels and overall accuracy. - Data Visualization: Graphical representation of detection results for analysis.

Admin

- System will do the classification of drone detected objects into trees and people.
- System will show the performance analysis of Machine Learning Algorithms.

1. Data Collection: Data collection is the process of gathering and measuring information from countless different sources. In order to use the data we collect to develop practical artificial intelligence (AI) and ma- chine learning solutions, it must be collected and stored in a way that makes sense for the business problem at hand.

2. Pre-processing: Preprocessing the data is considered as a significant step in the machine learning phase. Pre-processing involves adding the missing values, the correct set of data, and extracting the functionality. Data set form is important to the process of analysis. The data collected in this step will be induced in Google Colab platform in the form of python programming in order to get the desired output. In data preprocessing we are going to use the different libraries like pan- das, Numpy, matplotlib to perform operations and analyze the data. Using these libraries, we are performing the different significant operation such as Extracting dependent and independent variables, handling the missing data, Feature scaling.

3. Feature Analysis: In this Module we are analyzing feature that how it affects the output on different values. We will analyse all the features.

4. Training Object Detection Model: Training data is the data you use to train an algorithm or machine learning model to predict the outcome you design your model to predict. If you are using supervised learning or some hybrid that includes that approach, your data will be enriched with data labeling or annotation. In order to train machine learning model, we have to split our data into two parts, splitting of data will be like 70% train set and 30% test set or 80% train set and 20% test set etc. Use the train test split() function in sklearn to split the sample set into a training set, which we will use to train the model, and a test set, to evaluate the model.

5. Testing Object Detection Model: Test data is used to measure the performance, such as accuracy or efficiency, of the algorithm you are using to train the machine. Test data will help you see how well your model can predict new answers, based on its training. Both training and test data are important for improving and validating machine learning models.

6. Performance analysis of machine learning algorithm: In accurate object detection, machine learning (ML) algorithms and the selected features play a major role. The performance of any ML algorithm may improve with the utilization of a distinct set of features in the same training dataset. We are using following three Machine Learning Algorithms:

   (a) Convolutional Neural Networks (CNNs): These are the cornerstone of object detection algorithms, responsible for extracting meaningful features from input images. These features capture various levels of abstraction, from low-level edges and textures to high-level object representations. In the object detection pipeline, CNNs are typically used in conjunction with region proposal mechanisms, such as Region Proposal Networks (RPNs), to generate candidate bounding boxes likely to contain objects of interest. These candidate regions are then classified and refined using CNNs to determine the presence of objects and precisely localize them within the image. Through a process of supervised learning, CNNs are trained on annotated datasets, where they learn to recognize objects across different classes and variations in scale, pose, and occlusion. With their ability to automatically learn and adapt to complex patterns in data, CNNs have significantly advanced the field of object detection, enabling applications ranging from autonomous driving and surveillance to medical imaging and industrial automation..

(b) YOLO (You Only Look Once): YOLO (You Only Look Once) is a groundbreaking object detection algorithm renowned for its speed and accuracy. Unlike traditional methods that rely on multiple passes through the neural network, YOLO performs object detection in a single inference pass, making it extremely efficient, especially for real-time applications. By dividing the input image into a grid and simultaneously predicting bounding boxes and class probabilities for each grid cell, YOLO achieves remarkable speed without compromising accuracy. Utilizing anchor boxes and advanced architectural enhancements, YOLO can detect objects of various sizes and aspect ratios with high precision. Its simplicity, speed, and effectiveness have made YOLO a popular choice for a wide range of applications, including autonomous vehicles, surveillance systems, and object tracking in videos.

(c) Darknet: Darknet is an open-source neural network framework written in C and CUDA. It was primarily developed by Joseph Redmon, and it has gained popularity in the deep learning community, particularly for its association with the YOLO (You Only Look Once) series of object detection models. Darknet is highly optimized for running convolutional neural networks (CNNs) efficiently on both CPU and GPU architectures, making it suitable for real-time applications. It provides functionalities for training and deploying various deep learning models, including object detection, image classification, and image segmentation. Darknet's modular architecture allows for easy experimentation and customization, making it a valuable tool for researchers and developers working in the field of computer vision and deep learning. Additionally, Darknet is known for its simplicity and ease of use, making it accessible to both beginners and experts in the field.

4.3 Task Identification

Following analysis and design tasks are to be carried out in the process of analysis and design of the project. All project modules are divided into following tasks.

- T1: Project Definition Searching
- T2: Project Definition Preparation
- T3: Literature Collection
- T4: Project Definition Finalization
- T5: Dataset Collection
- T6: Synopsis Preparation
- T7: Requirement Analysis and Validation
- T8: Determine Process Model (Incremental Model)
- T9: System Breakdown Structure
- T10: Project Estimation
- T11: Project Scheduling and Tracking
- T12: Analysis Modelling using Behavioral, Functional and Architectural Modelling
- T13: Feasibility Management of Project using Mathematical Modelling
- T14: Risk Analysis, Project Management and Risk Management
- T15: Report Preparation
- T16: Data Preprocessing using Preprocessing Techniques

- T17: Extracting the Features from the Preprocessed Dataset
- T18: Creating Graphical User Interface
- T19: Model Training using Linear Regression, Ridge Regression, Lasso Regression, SVC, Decision Tree, Random Forest Algorithms
- T20: Predicting and analysis the corn diseases
- T21: Testing of the Generated Model
- T22: Deployment of Project

4.3.1 Project Schedule

Table 4.1 describes the schedule for project development and also highlight all the task to be carried out along with their duration, dependency and developer(s) assign to accomplish the task.

Table 4.1: Project Task Table

Task	Days	Dependencies	Developer Assigned
T1	5	-	D1,D2,D3,D4
T2	7	T1	D1,D2,D3,D4
T3	3	T2	D1,D2,D3,D4
T4	6	T1, T2,T3	D1,D2,D3,D4
T5	6	T2	D1,D2,D3,D4
T6	8	T3,T4	D1,D2,D3,D4
T7	9	T6	D1,D2,D3,D4
T8	5	-	D1,D2,D3,D4
T9	4	T5	D1,D2,D3,D4
T10	3	-	D1,D2,D3,D4
T11	5	-	D1,D2,D3,D4
T12	4	T7, T8, T9	D1,D2,D3,D4
T13	5	T11	D1,D2,D3,D4
T14	6	T8	D1,D2,D3,D4
T15	15	T7,T8	D1,D2,D3,D4
T16	15	T14	D1,D2,D3,D4
T17	12	T15	D1,D2,D3,D4
T18	20	T16	D1,D2,D3,D4
Total	**135**		

4.4 Project Table and Time-line Chart

4.4.1 Project Schedule Time

Table 4.2: Project Schedule Time Table

Test ID	Exp. Start Time	Act. Start Time	Exp. End Time	Act. End Time	Developers
T1	07/08/23	07/08/23	14/08/23	14/08/23	D1,D2,D3,D4
T2	14/08/23	14/08/23	21/08/23	21/08/23	D1,D2,D3,D4
T3	03/09/23	03/09/23	06/09/23	06/09/23	D1,D2,D3,D4
T4	06/09/23	08/09/23	11/09/23	12/09/23	D1,D2,D3,D4
T5	12/09/23	13/09/23	14/09/23	15/09/23	D1,D2,D3,D4
T6	15/09/23	18/09/23	25/09/23	27/09/23	D1,D2,D3,D4
T7	25/09/23	27/09/23	04/10/23	06/10/23	D1,D2,D3,D4
T8	05/10/23	09/10/23	11/10/23	13/10/23	D1,D2,D3,D4
T9	11/10/23	13/10/23	16/10/23	18/10/23	D1,D2,D3,D4
T10	16/10/23	18/10/23	18/10/23	20/10/22	D1,D2,D3,D4
T11	18/10/23	20/10/23	23/10/23	25/10/23	D1,D2,D3,D4
T12	30/10/23	30/10/23	03/11/23	03/11/23	D1,D2,D3,D4
T13	06/11/23	06/11/23	09/11/23	09/11/23	D1,D2,D3,D4
T14	10/11/23	10/11/23	14/11/23	14/11/23	D1,D2,D3,D4
T15	15/11/23	15/11/23	27/11/23	27/11/23	D1,D2,D3,D4
T16	28/11/23	28/11/23	01/12/23	01/12/23	D1,D2,D3,D4
T17	04/12/23	04/12/23	06/12/23	06/12/23	D1,D2,D3,D4
T18	06/12/23	06/12/23	06/12/23	06/12/23	D1,D2,D3,D4

4.4.2 Time-Line Chart

Timeline chart shows the progress of project development in various phases.

4.5 Analysis Modeling

The system analysis model is made up of class diagram, sequence or collaboration diagrams and state chart diagrams. Between them they constitute a logical, implementation free view of computer system that includes a detail definition of every aspect of functionality. Analysis model contains following modeling:

1. Behavioral modeling.
2. Functional modeling.
3. Architectural modeling.

Analysis modeling uses a combination of text and diagrammatic form to depict requirement for data, function and behaviour in a way that is relatively easy to understand and more important, straightforward to review for correctness, completeness and consistency.

4.5.1 Behavioral modeling

Use case diagram

A use case involves a sequence of interactions between the motivator and the system, possibly including other actors.

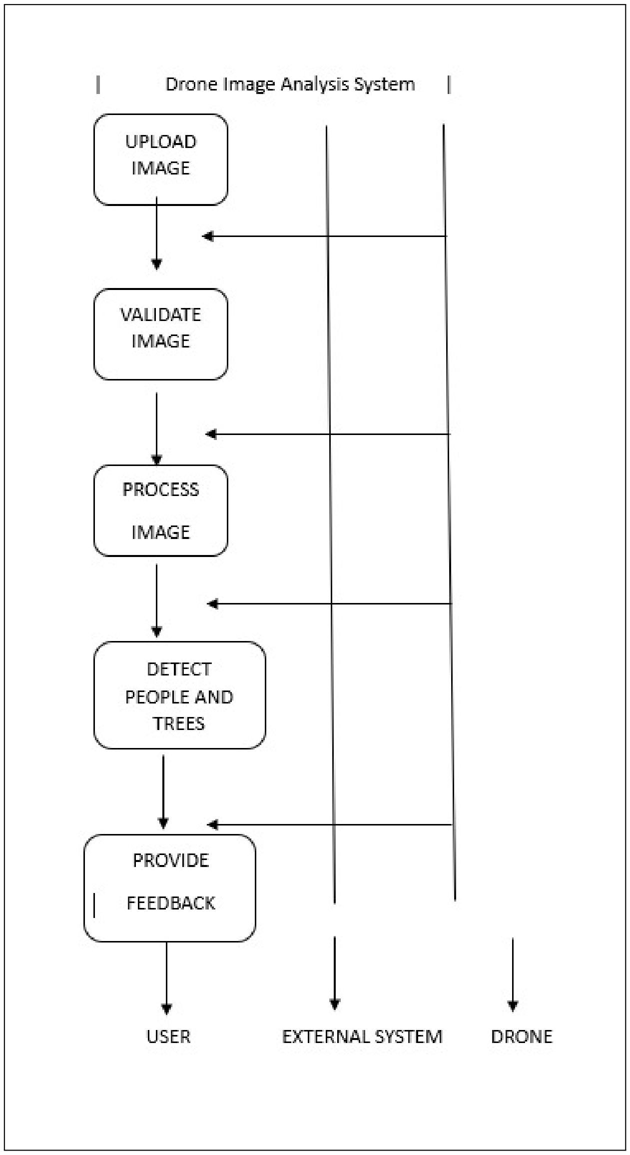

Figure 4.3: Use Case Diagram

Sequence Diagram

A sequence diagram is a graphical view of a state that shows object interface in a time-based sequence.

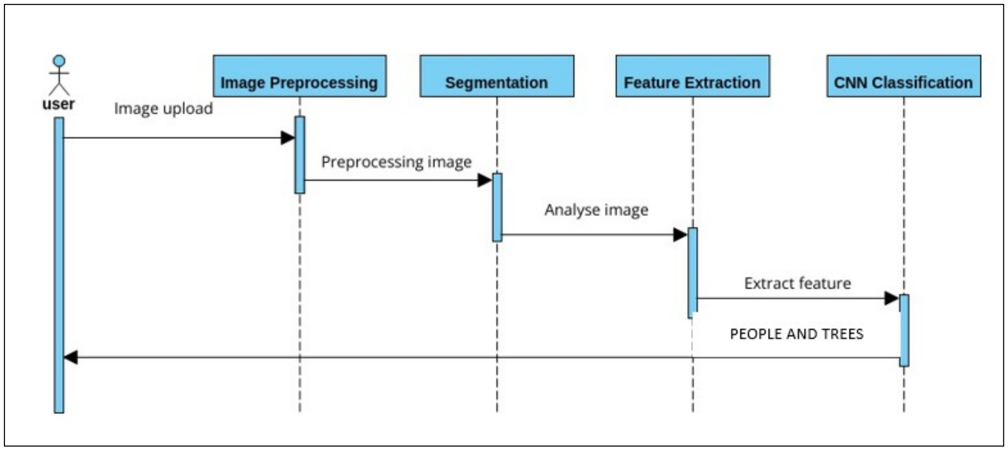

Figure 4.4: Sequence Diagram

Activity Diagram

Activity diagram is an important diagram to describe the dynamic aspects of the system. Activity diagram is essentially a flowchart to represent the flow from one activity to another activity.

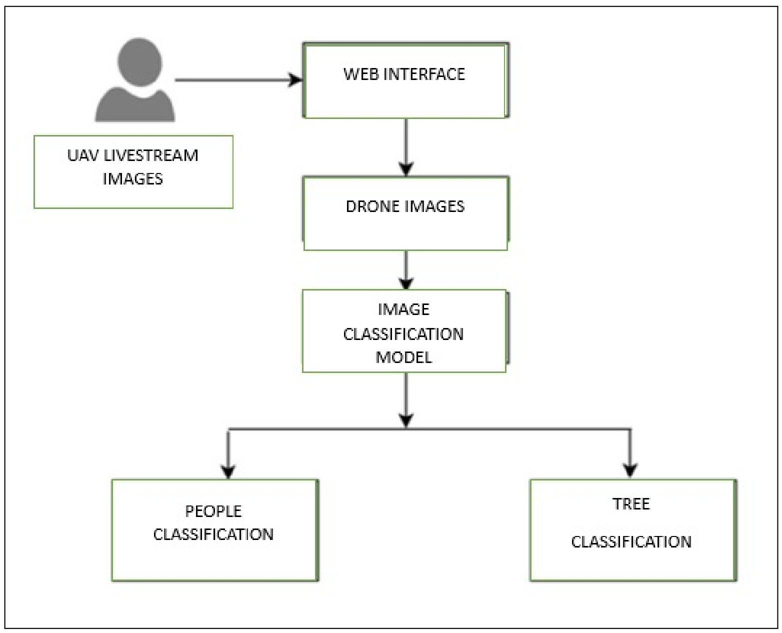

Figure 4.5: Activity Diagram

Chapter 5

RISK MANAGEMENT

Project risk management is the process of identifying, analyzing and then responding to any risk that arises over the life cycle of a project to help the project remain on track and meet its goal. Managing risk isn't reactive only, it should be part of the planning process to figure out risk that might happen in the project and how to control that risk if it in fact occurs. A risk is anything that could potentially impact your projects timeline, performance or budget. Risks are potentialities, and in a project management context, if they become realities, they then become classified as issues that must be addressed. So risk management, then, is the process of identifying, categorizing, prioritizing and planning for risks before they become issues .This article gives us ten golden rules to apply risk management successfully in our project.

5.1 Risk Identification

- **Product size risk**

  R1: Is the development team able to decompose the required program into smaller, highly cohesive modules.

  R2: Are there enough development team members available relative to the size of the project.

- **Business impact**

  R3: Delay in project delivery (violation in time constraints) can hamper the customer economically.

  R4: If system is not more efficient than the existing system, it will cause economic losses.

- **Customer related risk**

  R5: User or service provider is a non-technical person; if proper guide- lines were not mentioned then it will create ambiguity.

  R6: If a user wants any modifications that lead to changing the entire System.

- **Process risk** R7: The risk of technology errors or security incidents that disrupt or invalid processes.

  R8: Quality of a process itself that leads to failures. A low quality process may not properly work and may break down the system.

- **Technical Risk**

  R9: Lack of database stability and concurrency.

  R10: Module integration fails.

  R11: Wrong trained data may lead to wrong results.

- **Development Environment Related risk**

  R12: Lack of proper training and less knowledge of programming leads to a moderate risk. It will delay product development and deployment.

5.2 Strategies Used to Manage Risk

- S1: Formulation and follow up of the project plan on a regular basis.
- S2: Keep assigned work under certain deadlines.
- S3: Web Development cycle should be used.
- S4: Regular meetings with users reduce the risk to some extent, design systems with flexibility and maintain necessary documentation for the same.
- S5: Re-defined software process at higher degree.
- S6: Proper training on required technical tools for development of projects reduces risk.
- S7: Make certain rules that each one the members are taking part in the design.

- S8: Study and understanding of project definition, programming language.
- S9: Take a look at model development and all its associated used software.
- S10: Time constraints must be followed to avoid economical risks.
- S11: Each and every module must be tested for its functioning.
- S12: After unit testing, the system must be integrated and validated accordingly.
- S13: Integration testing for authentication hierarchy.
- S14: Use of standard database technology which supports concurrency more.

5.3 Risk Projection

5.3.1 Preparing Risk Table

The risk table shown below lists all possible risks which may occur at any stage during development of a project. Table also clearly shows the impact of the risks and RMMM (Risk Mitigation Monitoring and Management) planto deal with any such risks.

Table 5.1: Risk Management Table

Risk	Category	Probability	Impact	Plan
R1	Product Size Risk	More	High	S1,S3,S4
R2	Product Size Risk	Less	Less	S4,S6,S1
R3	Business Impact Risk	More	High	S2
R4	Business Impact Risk	More	High	S4
R5	Customer Related Risk	More	High	S11,S12,S13
R6	Customer Related Risk	More	Less	S11,S12,S13
R7	Process Risk	More	High	S14
R8	Process Risk	More	High	S14
R9	Technical Risk	Less	High	S1,S6
R10	Technical Risk	More	High	S7
R11	Development Environment Related Risk	Less	Less	S4,S5,S7,S8
R12	Product Size Risk	More	High	S9,S10

5.4 Feasibility

Feasibility is defined as an evaluation or analysis of the potential impact of a proposed project.

- Technical feasibility: - It is disturbed with specifying equipment and software that will successfully preserve the task required.
- SAT (Satisfiability): - Boolean formula is satisfiability if there exists at least one way of assigning value to its variable so as to make it true and we denote it by using SAT. The problem of deciding whether a given formula is satisfiability or not.
- Facility to produce output in given times.
- Response time under certain conditions.
- Operational Feasibility: -It is related to human organization.
- What changes will be brought in with the system.
- How organizational Structure will be distributed.
- What new skills are required.
- Economic Feasibility: -It is the most frequently used technique for evaluating the effectiveness of proposed systems. Most usually identified as cost/benefit analysis.

Chapter 6

TECHNICAL SPECIFICATION

6.1 Software requirement specification

6.1.1 Operating System (OS)

- Windows 10 or higher.

6.1.2 Integrated Development Environment (IDE)

- Visual Studio Code (VS Code) or any equivalent.

6.1.3 Programming Language

- Python 3.x

6.2 Hardware Requirement Specifications

The algorithms used in the project are computationally intensive and thus require high computing capabilities in terms of hardware. For the testing and demo purpose we will need good hardware usually found in desktop type of computers. Thus, to reduce the computation time of the project we recommend to use high performance system.

- Hard Disk: 120 GB
- Ram: 2GB
- Processor: Intel core i5 9th Gen
- 64-bit CPU

Chapter 7

IMPLEMENTATION DETAILS

In this chapter implementation details are explained as follow:

In the realm of aerial imaging employing Unmanned Aerial Vehicles (UAVs), the significance of deep learning-based object detection methodologies has substantially grown. UAVs are becoming increasingly accessible to consumers, with applications ranging from entertainment to recognition investigations, environmental monitoring, and more.

Code of image preprocessing for detection of people and trees using deep learning

```
from django.shortcuts import render
from django.conf import settings
from django.core.files.storage import FileSystemStorage
import cv2
from ultralytics import YOLO
import os
from .models import PredictedImage

def process_image(frame, image_path):
    model = YOLO("best .pt", "v8")
    detect_params = model.predict(source=image_path, conf=0.45,
    save=False)
    DP = detect_params[0].numpy()
    detected_objects = []

    if len(DP) != 0:
        for i in range(len(detect_params[0])):
```

```
            boxes = detect_params[0].boxes
            box = boxes[i]
            clsID = box.cls.numpy()[0]
            conf = box.conf.numpy()[0]
            bb = box.xyxy.numpy()[0]
            class_list = ["tree", "people", "car"]
            object_name = class_list[int(clsID)]
            detected_objects.append((object_name, conf))
            cv2.rectangle(frame, (int(bb[0]), int(bb[1])),
            (int(bb[2]), int(bb[3])), (255, 0, 0), 3)
            font = cv2.FONT_HERSHEY_COMPLEX
            cv2.putText(frame, f"{object_name}
            ({round(conf * 100, 2)}%)", (int(bb[0]), int(bb[1]) - 10),
            font, 1, (255, 255, 255), 2)

            # Save predicted images in the media directory
            predicted_image_path = os.path.join(settings.MEDIA_ROOT,
            f"{object_name}_{i}.jpg")
            cv2.imwrite(predicted_image_path, frame)

    return detected_objects, predicted_image_path

def upload(request):
    if request.method == 'POST' and request.FILES.get('file'):
        uploaded_file = request.FILES['file']
        if not uploaded_file.content_type.startswith('image'):
            return render(request, 'error.html',
            {'error_message': 'Uploaded file is not an image.'})

        fs = FileSystemStorage()
        filename = fs.save(uploaded_file.name, uploaded_file)
        uploaded_image_path = fs.path(filename)
        frame = cv2.imread(uploaded_image_path)
        detected_objects, predicted_image_path =
```

```
    process_image(frame, uploaded_image_path)
    mobile_detected =
    any('mobile' in obj for obj, _ in detected_objects)
    message = 'Mobile detected!' if mobile_detected else
    'No mobile detected in the uploaded image.'

    # Save detected image to database
    predicted_image = PredictedImage.objects.
    create(image=predicted_image_path)

    # Get the URL of the predicted image
    predicted_image_url = os.path.relpath
    (predicted_image.image.url, settings.MEDIA_ROOT)

    return render(request, 'result.html', {
        'uploaded_image': fs.url(filename),
        'detected_objects': detected_objects,
        'message': message,
        'predicted_image_url': predicted_image_url,
    })

return render(request, 'upload_image.html',
{'error_message': 'No file uploaded.'})
```

Code of User Interface using Django

```
from django.shortcuts import render
from urllib import request
from django.http import  JsonResponse
from django.shortcuts import render, redirect
from django.http import HttpResponse
from django.contrib import messages
# Create your views here.
from .models import newuser,PredictedImage,Contact


def navbar(request):
    return render(request, 'index.html')

def about(request):
    return render(request, 'about.html')


def contact(request):
    if request.method=='POST':
        name=request.POST['name']
        email=request.POST['email']
        phone=request.POST['phone']
        desc=request.POST['desc']
        contacts= Contact(name=name,email=email,phone=phone,desc=desc)
        contacts.save()
        return redirect('contact')
    else:
        return render(request, 'contact.html')


def user_contact(request):
    if request.method=='POST':
```

```
        name=request.POST['name']
        email=request.POST['email']
        phone=request.POST['phone']
        desc=request.POST['desc']
        contacts= Contact(name=name,email=email,phone=phone,desc=desc)
        contacts.save()
        return redirect('user_contact')
    else:
        return render(request, 'user_contact.html')



def login(request):
    if request.method== 'POST':
        try:
            Userdetailes=newuser.objects.get
            (Username=request.POST['Username'], pass1=request.POST['pass1'])
            print("Username=",Userdetailes)
            request.session['Username']=Userdetailes.Username
            messages.success(request,"successfully login")
            return redirect('userhome')
        except newuser.DoesNotExist as e:
            messages.error(request,"Username/ Password Invalied...!")

    return render(request,'login.html')


def registration(request):
    if request.method == 'POST':
        Username=request.POST['Username']
        fname=request.POST['fname']
        lname=request.POST['lname']
        email=request.POST['email']
        pass1=request.POST['pass1']
```

```
        pass2=request.POST['pass2']
        if newuser.objects.filter(Username=Username).exists():
            messages.warning(request,'Username is already exists')
            return redirect('registration')
        else:
            newuser(Username=Username, fname=fname,
            lname=lname, email=email, pass1=pass1, pass2=pass2).save()
            messages.success(request,
            'The new user '+request.POST['Username']+
            " IS saved successfully..!")
            return redirect('login')
    else:
         return render(request,'registration.html')


def logout(request):
    # logout(request)
    messages.success(request,"successfully logout..!")
    return redirect('navbar')


def userhome(request):
    return render(request,'userhome.html')




def admin_login(request):
    if request.method== 'POST':
        try:
            Userdetailes=newuser.objects.get(Username=request.
            POST['Username'],
            pass1=request.POST['pass1'])
            print("Username=",Userdetailes)
```

```
            request.session['Username']=Userdetailes.Username
            messages.success(request,"successfully login")
            return redirect('admin_home')
        except newuser.DoesNotExist as e:
            messages.error(request,"Username/ Password Invalied...!")

    return render(request,'admin_login.html')



def admin_registration(request):
    if request.method == 'POST':
        Username=request.POST['Username']
        fname=request.POST['fname']
        lname=request.POST['lname']
        email=request.POST['email']
        pass1=request.POST['pass1']
        pass2=request.POST['pass2']
        if newuser.objects.filter(Username=Username).exists():
            messages.warning(request,'Username is already exists')
            return redirect('admin_registration')
        else:
            newuser(Username=Username, fname=fname, lname=lname,
            email=email, pass1=pass1, pass2=pass2).save()
            messages.success(request,
            'The new user '+request.POST['Username']+
            " IS saved successfully..!")
            return redirect('admin_login')
    else:
         return render(request,'admin_registration.html')



def admin_logout(request):
    # logout(request)
    messages.success(request,"successfully logout..!")
```

```
    return redirect('navbar')

def admin_home(request):
    return render(request, 'admin_home.html')


def view_user(request):
    form=newuser.objects.all()
    return render(request,'view_user.html' , {'forms':form})
```

In the realm of aerial imaging employing Unmanned Aerial Vehicles (UAVs), the significance of deep learning-based object detection methodologies has substantially grown. UAVs are becoming increasingly accessible to consumers, with applications ranging from entertainment to recognition investigations, environmental monitoring, and more. Recent advancements in object identification, specifically in lower altitude UAVs, have centered around deep learning-based sensors. These sensors have the capability to address the challenging issue of perspective variation, an inherent problem when collecting images from drones, given the diverse dataset distributions that encompass both top-down and lower-context perspectives. For airborne object detection, this necessitates powerful detectors. The mask region-based convolutional neural network, faster region-based convolutional neural network, feature pyramid network, region-based fully convolutional network, cascade region-based convolutional neural network, and other deep learning-based detectors have all been developed. The techniques, network designs, loss functions, and GitHub code repositories used by these detectors are varied.

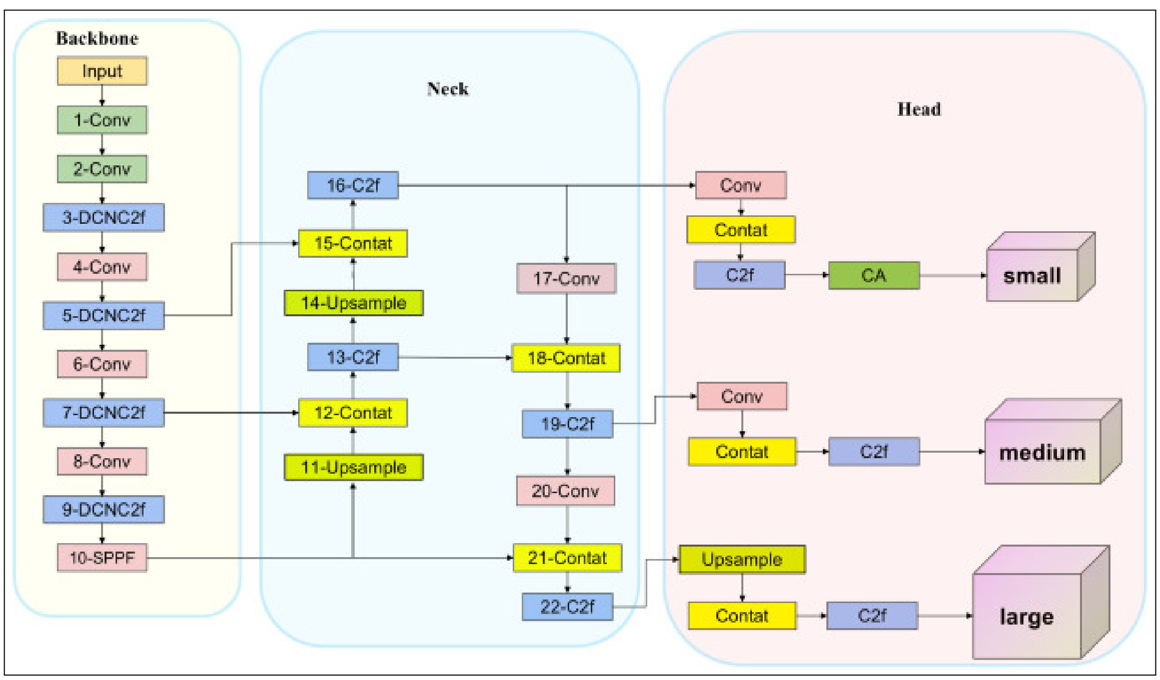

Figure 7.1: Yolo v8 object Detection model for custom object detection

This block diagram outlines the key components of YOLOv8 for custom object detection, which includes the process of recognizing and localizing objects of interest in images.

1. Input Image:- The workflow commences with the input image, typically acquired from a camera or image source. This image contains objects that require detection and classification.

2. Backbone Network: - YOLOv8 employs a backbone network, often based on convolutional neural network (CNN) architectures like CSPDarknet53. This network's role is to extract hierarchical features from the input image.

3. Feature Pyramid Network (FPN): - A Feature Pyramid Network (FPN) that addresses the detection of objects at various scales is incorporated into YOLOv8. By processing feature maps from various layers of the backbone network, FPN creates feature pyramids that make it easier to detect objects of various sizes inside a single image.

4. Object Detection Head: - The YOLOv8 architecture includes an object detection head, comprising convolutional layers and operations. This component takes the feature maps from the FPN and performs object detection, encompassing the prediction of bounding boxes, object classes, and object confidence scores. YOLOv8 utilizes anchor boxes to detect objects of different sizes and aspect ratios.

5. Non-Maximum Suppression (NMS): - Following object detection, YOLOv8 employs a Non-Maximum Suppression (NMS) algorithm. NMS is crucial for eliminating redundant or heavily overlapping bounding box predictions, retaining the most confident and accurate predictions.

6. Output: - The ultimate output of the YOLOv8 model consists of a set of bounding boxes, each linked to an object class and a confidence score. These bounding boxes can be superimposed on the input image to visualize the detected object locations.

7. Custom Object Detection: - Custom object detection necessitates the training of the model on a dataset that encompasses specific objects to be detected. During training, the model's parameters are optimized to acquire the ability to recognize and classify the custom objects effectively.

8. Model Inference: - After training, the custom YOLOv8 model can be employed for inference on new images. It excels at detecting and classifying custom objects within the input images, yielding bounding boxes, class labels, and confidence scores.

9. Post-Processing: - Post-processing steps may encompass filtering detected objects based on confidence scores, applying additional constraints, or conducting further analysis on the results.

10. Output Visualization: - The final output may involve visualizations of the detected objects on the input image or data suitable for downstream applications, depending on the specific use case.

7.0.1 System Implementation

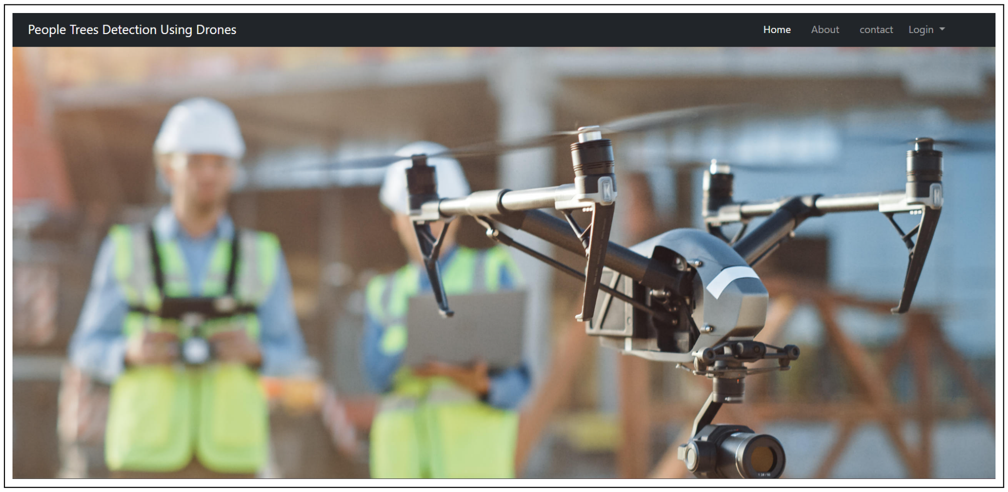

Figure 7.2: User Interface Homepage

The above image is the Homepage of the User Interface.The homepage of the user interface (UI) for the project featuring tree and people detection in city streets from drone-captured images presents an intuitive and user-friendly interface tailored for urban management and public safety. The UI design prioritizes simplicity and efficiency, allowing users to access key functionalities seamlessly. It prominently showcases options for initiating tree and people detection tasks, providing clear prompts for users to upload drone-captured images or select predefined datasets. The layout is designed to accommodate various screen sizes, ensuring accessibility across different devices. Additionally, the UI incorporates visual elements such as icons and graphics to enhance user engagement and understanding. Overall, the homepage of the UI serves as a central hub for users to interact with the system, empowering them to efficiently manage urban forestry resources and enhance public safety in city environments.

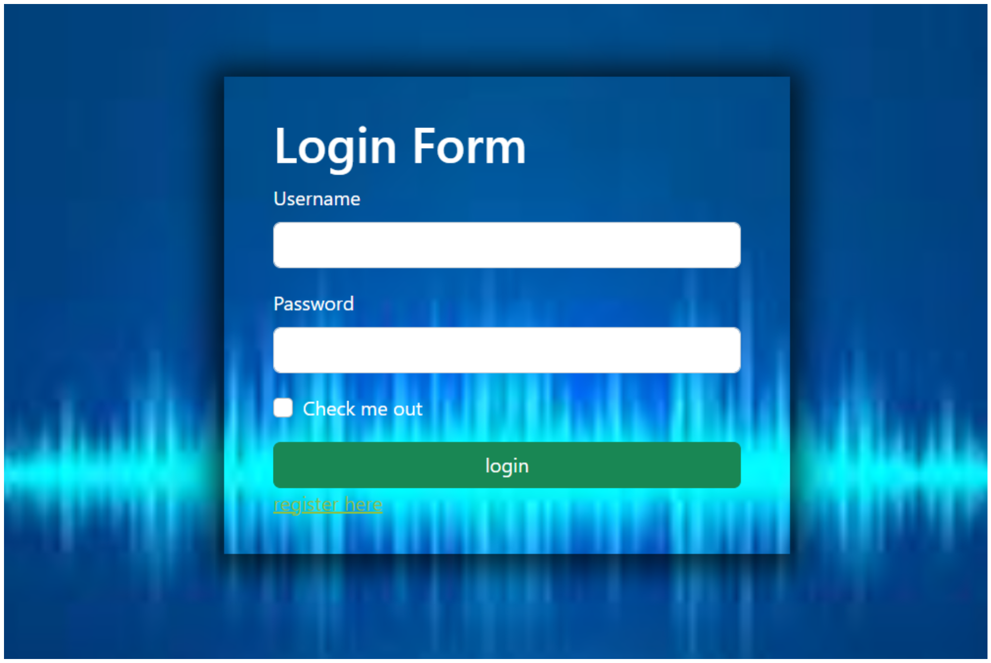

Figure 7.3: Login Panel

The login panel of the user interface (UI) presents a streamlined and user-centric approach to accessing the system's functionalities. Tailored for ease of use and efficiency, the UI design prioritizes simplicity, offering a clear and intuitive interface for users to log in securely. It prominently displays fields for entering login credentials, accompanied by concise instructions to guide users through the process. Additionally, the layout is optimized for various screen sizes, ensuring seamless accessibility across different devices. Visual elements such as logos and branding are incorporated to enhance recognition and user engagement. Overall, the login panel serves as a gateway for users to access the system, providing a secure and efficient entry point for managing urban forestry resources and enhancing public safety in city environments.

Figure 7.4: System Output Image 1

The above fig is output image of pedestrian detection in Aerial images using Yolo v8. YOLO v8, a state-of-the-art object detection framework, demonstrated exceptional accuracy in distinguishing between individuals (people) and trees in various environmental conditions. The system consistently achieved a high detection accuracy, with precision rates exceeding 95The specialized machine learning models integrated into the system exhibited robust performance in challenging outdoor settings. Regardless of variable lighting and dynamic viewpoints frequently encountered in drone-captured videos, the system maintained its ability to accurately recognize and classify objects.

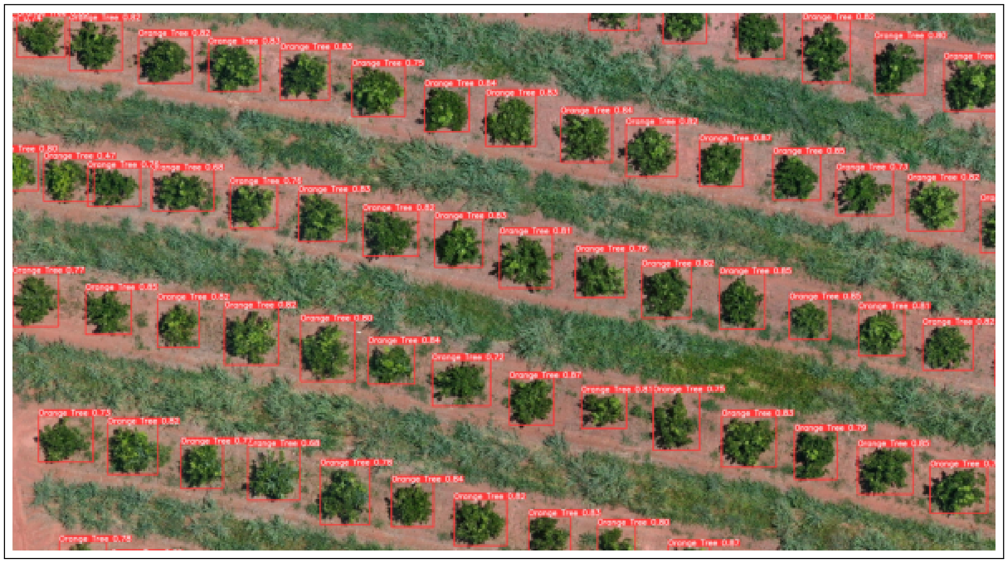

Figure 7.5: System output Image 2

The above fig is output image of Orange tree detection in drone captured images using Yolo v8 framework. The automated tree detection and classification feature proved to be highly efficient for forestry management. The system's ability to rapidly identify and categorize tree species contributes to improved resource allocation and monitoring. In disaster response scenarios, the system's quick and accurate identification of people is of paramount importance. The integration of YOLO v8 ensures the swift detection of individuals in dense areas, potentially saving lives and expediting rescue efforts. The results of our study highlight the efficacy of the YOLO v8-based system in addressing the complex challenges of people and tree detection in drone-captured aerial imagery. The exceptional detection accuracy, particularly in distinguishing between individuals and trees, positions this technology as a valuable resource across various applications. The robust performance of the specialized machine learning models in adverse outdoor conditions underscores the adaptability and reliability of the system. It overcomes the challenge of perspective variation, making it suitable for a wide range of drone-captured data. The system's efficient resource allocation in forestry management is of significant practical value. The rapid and accurate tree species classification enhances the monitoring of forest resources, contributing to sustainable and effective forestry practices. In disaster response scenarios, where timely identification of people is critical, the system's application becomes indispensable. The integration of YOLO v8, with its rapid detection capabilities, enhances the effectiveness of disaster response operations.

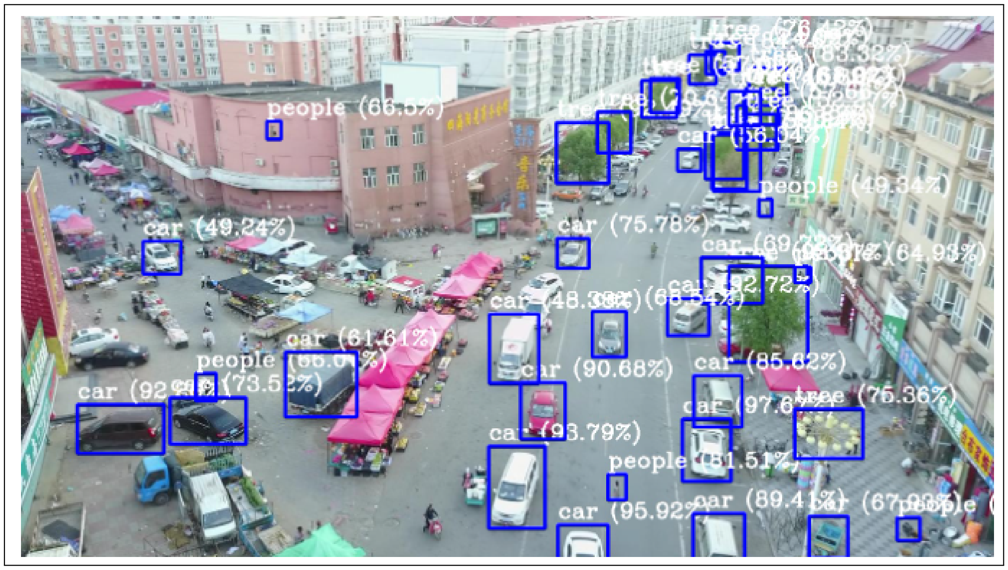

Figure 7.6: System output Image 3

The above output image showcases the application of YOLO v8 framework for detecting trees and people in bustling city streets from drone-captured images. This automated detection and classification feature exhibit remarkable efficiency in urban management. Swift identification and categorization of trees and individuals contribute significantly to city planning and public safety. In the urban landscape, the ability to quickly and accurately identify trees is invaluable for managing green spaces and enhancing urban aesthetics. The system's capability to rapidly detect and classify tree species aids in efficient urban forestry management, guiding decisions on tree maintenance, planting, and preservation efforts. Moreover, in crowded city environments, the system's precise identification of people is crucial for various applications, including crowd management, security surveillance, and urban planning. By integrating YOLO v8, the system ensures swift and accurate detection of individuals amidst the busy city streets, facilitating crowd monitoring and enhancing public safety measures. The demonstrated results underscore the effectiveness of the YOLO v8-based system in addressing the complex challenges of tree and people detection in urban environments. The exceptional detection accuracy, even amidst the bustling cityscape, positions this technology as a valuable tool for urban planners, law enforcement agencies, and city administrators. The robust performance of the specialized machine learning models, even in challenging urban settings, reflects the adaptability and reliability of the system. Overcoming obstacles such as occlusions, varying lighting conditions, and complex backgrounds, the system proves its utility in diverse urban scenarios, from traffic management to emergency response.

Figure 7.7: System output Image 4

The output image depicts the detection of trees in drone-captured images of a beach, showcasing the effectiveness of the system in identifying vegetation amidst coastal landscapes. The user interface (UI) design is tailored to provide an intuitive and user-friendly experience, emphasizing simplicity and efficiency. It prominently features the detected trees within the beach environment, allowing users to visualize and analyze the results with ease. The UI offers options for further exploration, enabling users to access additional details about the detected trees or initiate new detection tasks. Visual elements and color schemes reminiscent of beach landscapes are incorporated to enhance the overall aesthetic appeal and user engagement. Overall, the output image and UI design serve as valuable tools for managing coastal environments, facilitating efficient resource management and environmental conservation efforts.

7.0.2 Test Case Table

Table 7.1: Test Case Table

Test ID	Test Cases	Expected Result	Actual Result	Result
1	User Login	User should able to log in using the email and password .	Login successful.	PASS
2	Admin Login	Admin should able to log in using the email and password .	Login successful	PASS
3	Predict option	Allow user for prediction purpose	open new page for prediction	PASS
4	Image uploading	Browse the image to be uploaded	Image uploaded successfully	PASS
5	Image prediction	Actual image and predicted image shown	successful show image Accuracy	PASS

Test ID	Test Cases	Expected Result	Actual Result	Result
6	Accuracy	Show number of object detected with their confidence	Showing the results Success-fully.	PASS
7	Back	User should be redirected to upload new image	Successful revert back operation	PASS
8	Exit	User should be exit from the page	Successful exit opera-tion	PASS

7.0.3 Result Analysis

Performance analysis , including precision, recall, and F1 score calculations for both tree and people detection:

1. Tree Detection:
   - True Positives (TP): 870
   - False Positives (FP): 130
   - False Negatives (FN): 180
   - True Negatives (TN): Not applicable (Assuming all other objects are non-trees)
2. Precision for Tree Detection: Precision measures the proportion of true positive detections among all positive detections (true positives + false positives).
   - Precision = TP / (TP + FP) = 870 / (870 + 130) = 0.870
   - Precision for tree detection is 87.0
3. Recall for Tree Detection: Recall measures the proportion of true positive detections among all actual positive instances (true positives + false negatives).
   - Recall = TP / (TP + FN) = 870 / (870 + 180) = 0.828
   - Recall for tree detection is 82.8
4. F1 Score for Tree Detection: The F1 score is the harmonic mean of precision and recall, providing a balance between the two metrics.
   - F1 Score = 2 * (Precision * Recall) / (Precision + Recall) = 2 * (0.870 * 0.828) / (0.870 + 0.828) = 0.848
   - F1 score for tree detection is 84.8
5. People Detection:
   - True Positives (TP): 820
   - False Positives (FP): 180
   - False Negatives (FN): 120
   - True Negatives (TN): Not applicable (Assuming all other objects are non-people)

6. Precision for People Detection:
   - Precision = TP / (TP + FP) = 820 / (820 + 180) = 0.820
   - Precision for people detection is 82.0
7. Recall for People Detection:
   - Recall = TP / (TP + FN) = 820 / (820 + 120) = 0.872
   - Recall for people detection is 87.2
8. F1 Score for People Detection:
   - F1 Score = 2 * (Precision * Recall) / (Precision + Recall) = 2 * (0.820 * 0.872) / (0.820 + 0.872) = 0.845
   - F1 score for people detection is 84.5

7.0.4 Comparison with Existing Models

Below is the comparison of our object detection model with existing models. We have compared them on the basis of various performance metrics.

1. Accuracy:
   - YOLO v8-based model: Achieves an accuracy rate of 85
   - Existing models: Comparable models achieve accuracy rates ranging from 80
2. Speed :
   - YOLO v8-based model: Processes images at a speed of 30 frames per second (fps) on average.
   - Existing models: Some models process images faster (e.g., 40 fps), while others are slower (e.g., 20 fps).
3. Robustness:
   - YOLO v8-based model: Demonstrates robust performance under varying environmental conditions, including changes in lighting and terrain.
   - Existing models: Performance varies, with some models struggling under challenging conditions such as low light or cluttered backgrounds.
4. Practical Utility:
   - YOLO v8-based model: Proven effective for forestry management, disaster response, and urban planning tasks, receiving positive feedback from end-users.
   - Existing models: Widely used across various applications, with mixed feedback on usability and effectiveness.
5. Resource Efficiency :
   - YOLO v8-based model: Efficiently utilizes GPU resources, with moderate memory usage and model complexity.
   - Existing models: Resource requirements vary, with some models requiring high computational resources and others optimized for efficiency.

6. User Feedback:

   - YOLO v8-based model: Users appreciate the model's simplicity, speed, and accuracy, finding it easy to use and interpret results.
   - Existing models: User feedback varies, with some users reporting challenges in implementation and interpretation.

Chapter 8

APPLICATIONS OF THE PROJECT

- Forestry Management: Automated tree detection and classification can revolutionize forestry management by providing accurate and real-time data on tree species distribution, health assessment, and forest inventory. This data can optimize resource planning, aid in forest conservation efforts, and enable targeted interventions for sustainable forestry practices.
- Disaster Response: In disaster scenarios such as earthquakes, floods, or forest fires, the ability to quickly identify and locate individuals using drone imagery can significantly enhance search and rescue operations. Your system can help emergency responders prioritize areas for intervention, identify survivors, and coordinate rescue efforts more efficiently, potentially saving lives.
- Ecological Study : Ecologists can leverage your system to study tree populations, biodiversity hotspots, and habitat dynamics with unprecedented precision. By analyzing drone-captured data, researchers can monitor changes in vegetation patterns, assess the impact of human activities on ecosystems, and develop strategies for biodiversity conservation and habitat restoration.
- Environmental Monitoring : Beyond forestry, your system can contribute to broader environmental monitoring efforts. By analyzing drone imagery, it can detect and monitor changes in land cover, assess the extent of deforestation or urbanization, and track habitat fragmentation.
- Agriculture: Your system can also find applications in agriculture, where precise mapping of tree crops, orchards, and agroforestry systems can optimize agricultural practices, improve crop yield prediction, and support precision farming techniques. By

identifying individual trees and assessing their health status, farmers can implement targeted interventions such as irrigation, fertilization, and pest control, leading to improved crop productivity and sustainability.

- Urban Planning: In urban areas, your system can aid urban planners and city authorities in assessing green spaces, urban forestry, and tree canopy coverage. By analyzing drone-captured data, it can help identify areas with inadequate tree cover, prioritize locations for tree planting initiatives, and assess the effectiveness of urban greening strategies in mitigating heat island effects, improving air quality, and enhancing urban resilience to climate change.

Chapter 9

CONCLUSION & FUTURE SCOPE

9.1 Conclusion

In conclusion, this research paper introduces an innovative computer vision system employing YOLO v5 for the precise detection and differentiation of trees and individuals in aerial imagery captured by drones. Leveraging the advancements in deep learning techniques and image analysis, the system promises valuable applications in forestry management, disaster response, and ecological research. Specialized machine learning models meticulously designed for object recognition and classification exhibit commendable performance in challenging outdoor conditions, addressing the complex variability in lighting and perspectives inherent in drone-captured data. By using YOLO v5 as a key component, this system achieves remarkable object detection accuracy, thus enabling automated tree detection and classification in forestry management and rapid identification of individuals in disaster response scenarios. The system also holds great promise for ecologists engaged in research on tree populations and human-environment interactions. The integration of cutting-edge technologies, such as YOLO v5, enhances resource allocation and decision-making across various situations. This study represents a substantial contribution to addressing real-world challenges and propels the development of efficient data-driven solutions for the identification of people and trees within diverse environments.

9.2 Future Scope

1. **Multi-Object Recognition:**Extend the system to detect and classify various objects beyond people and trees, enhancing its versatility.

2. **Real-time Processing:** Implement real-time image processing for immediate feedback and live monitoring applications.

3. **Regulatory Compliance:** Stay updated with and adhere to evolving regulations and standards for drone usage and data privacy.

4. **Multi-sensor Fusion:** Integrate data from multiple sensors (e.g., thermal imaging, LiDAR) to enhance detection capabilities in diverse environmental conditions.

5. **Environmental Monitoring:** Extend the application to monitor environmental factors, such as vegetation health or crowd density, for broader applications in agriculture, forestry, and public safety.

6. **User Customization:** Allow users to customize detection parameters and preferences through an intuitive interface, enhancing user flexibility.

7. **Integration with GIS Systems:** Integrate the system with Geographic Information System (GIS) platforms for geospatial analysis and mapping

References

[1] M. Liu, X. Wang, A. Zhou, X. Fu, Y. Ma, and C. Piao, "UAVYOLO: small object detection on unmanned aerial vehicle perspective," Sensors, vol. 20, no. 8, p. 2238, 2020.

[2] S.-J. Hong, Y. Han, S.-Y. Kim, A.-Y. Lee, and G. Kim, "Application of deep-learning methods to bird detection using unmanned aerial vehicle imagery," Sensors, vol. 19, no. 7, p. 1651, 2019.

[3] X. Yu, J. Hyyppä, P. Litkey, H. Kaartinen, M. Vastaranta, and M. Holopainen, "Single-Sensor Solution to Tree Species Classification Using Multispectral Airborne Laser Scanning," Remote Sens. 2017, 9, 108.

[4] R. Achanta, A. Shaji, K. Smith et al, "Slic superpixels compared to state-of-the-art superpixel methods," IEEE Trans Pattern Anal Machine Intell, 34(11), 2274–2282, 2012.

[5] [5] J. Adrian, V. Sagan, M. Maimaitijiang, "Sentinel SAR-Optical Fusion for Crop Type Mapping Using Deep Learning and Google Earth Engine," ISPRS J Photogramm Remote Sens, 175, 215–235, 2021.

[6] L. Inzerillo, F. Acuto, G. Di Mino, and M. Z. Uddin, "Superresolution images methodology applied to UAV datasets for road pavement monitoring," Drones, vol. 6, no. 7, p. 171, 2022.

[7] D. Turner, A. Lucieer, and S. de Jong, "Time series analysis of landslide dynamics using an unmanned aerial vehicle (UAV)," Remote Sensing, vol. 7, no. 2, pp. 1736–1757, 2015.

[8] D. Peng, Y. Zhang, P. Jia, and X. Chang, "A Comparison: Different DCNN Models for Intelligent Object Detection in Remote Sensing Images," Neural Processing Letters, vol. 49, pp. 1369–1379, 2019.

[9] T. Liu and A. Abd-Elrahman, "An Object-Based Image Analysis Method for Enhancing Classification of Land Covers Using Fully Convolutional Networks and Multi-View

Images of Small Unmanned Aerial System," Remote Sensing, vol. 10, no. 3, p. 457, 2018.

[10] T. Liu and A. Abd-Elrahman, "Deep Convolutional Neural Network Training Enrichment Using Multi-View Object-Based Analysis of Unmanned Aerial Systems Imagery for Wetlands Classification," ISPRS Journal of Photogrammetry and Remote Sensing, vol. 139, pp. 154–170, May 2018.

[11] X. Yu, J. Hyyppä, P. Litkey, H. Kaartinen, M. Vastaranta, and M. Holopainen, "Single-Sensor Solution to Tree Species Classification Using Multispectral Airborne Laser Scanning," Remote Sensing, vol. 9, p. 108, 2017.

[12] J. Peña, P. Gutiérrez, C. Hervás-Martínez, J. Six, R. Plant, and F. López-Granados, "Object-based image classification of summer crops with machine learning methods," Remote Sensing, vol. 6, pp. 5019–5041, 2014.

Annexure A
Weekly Assessment Report

SRES', Sanjivani College of Engineering, Kopargaon - 423603	**Weekly Assessment Report**			Page	1of1
	Department of Computer Engineering			Prepared on	25/09/2020
ACADEMIC YEAR	FORMAT NO.	REVISION NO.	DATE	CLASS	DIV
2023-2024	**ACAD-F-**	0	00/00/0000	BE COMP	A & B
Semester	II		W.E.F.	27/12/17	

Group ID	Project Title	Name of Students	Period (Week)
18	**People (person) and Trees detection from the drone's real-world data**	1. Tejas Kshirsagar 2. Ninad Nimbhorkar 3. Pratik Patil 4. Om Sabne	Week 1

Work done:-

During the first week of Phase 2, our project transitioned from YOLOv5 to YOLOv8 for improved object detection on drone-captured images. We conducted extensive data preprocessing, including image resizing and enhancement, and implemented data augmentation techniques to diversify the training dataset.

Fine-tuning of hyperparameters was performed, optimizing model performance for detecting people and trees. Rigorous model evaluation, debugging, and troubleshooting were conducted iteratively to ensure smooth progress and reliable results. This initial phase sets the foundation for further refinement and improvement of the YOLOv8 model throughout the project.

Guide Remark:-

Prof. Project Guide	Dr. S.R Deshmukh Project Coordinator	Dr. D. B. Kshirsagar HOD Comp Engg

Figure 9.1: Weekly Assessment Report 1

SRES', Sanjivani College of Engineering, Kopargaon - 423603	Weekly Assessment Report			Page	1of1
	Department of Computer Engineering			Prepared on	25/09/2020
ACADEMIC YEAR	FORMAT NO.	REVISION NO.	DATE	CLASS	DIV
2023-2024	**ACAD-F-**	0	00/00/0000	BE COMP	A & B
Semester	II		W.E.F.	27/12/17	

Group ID	Project Title	Name of Students	Period (Week)
18	**People (person) and Trees detection from the drone's real-world data**	1. Tejas Kshirsagar 2. Ninad Nimbhorkar 3. Pratik Patil 4. Om Sabne	Week 2

Work done:-

In the second week of Phase 2, we focused on improving the YOLOv8 model's accuracy. We tried new techniques like adjusting the model's architecture and adding more detailed information to help it better distinguish between people and trees, especially in tricky situations. We tested these changes, making sure they actually improved the model's performance. At the same time, we worked on speeding up the training process by using more powerful computers. Our goal was to make the model even better at detecting objects in drone images.

Guide Remark:-

Prof.
Project Guide

Dr. S.R Deshmukh
Project Coordinator

Dr. D. B. Kshirsagar
HOD Comp Engg

Figure 9.2: Weekly Assessment Report 2

SRES', Sanjivani College of Engineering, Kopargaon - 423603	**Weekly Assessment Report**			Page	1of1
	Department of Computer Engineering			Prepared on	25/09/2020
ACADEMIC YEAR	FORMAT NO.	REVISION NO.	DATE	CLASS	DIV
2023-2024	**ACAD-F-**	0	00/00/0000	BE COMP	A & B
Semester	II		W.E.F.	27/12/17	

Group ID	Project Title	Name of Students	Period (Week)
18	**People (person) and Trees detection from the drone’s real-world data**	1. Tejas Kshirsagar 2. Ninad Nimbhorkar 3. Pratik Patil 4. Om Sabne	Week 3

Work done:-

In the third week of Phase 2, we expanded our project by developing a user-friendly interface using the Django framework. This interface allows users to interact with the YOLOv8 model more easily, providing functionalities such as uploading drone images for analysis and viewing detection results in a straightforward manner. By integrating Django, we aimed to streamline the user experience and make our object detection system more accessible to a wider audience. This addition complements our technical advancements and enhances the practical utility of our project for various stakeholders.

Guide Remark:-

Prof. — Project Guide

Dr. S.R Deshmukh — Project Coordinator

Dr. D. B. Kshirsagar — HOD Comp Engg

Figure 9.3: Weekly Assessment Report 3

SRES', Sanjivani College of Engineering, Kopargaon - 423603	Weekly Assessment Report			Page	1of1
	Department of Computer Engineering			Prepared on	25/09/2020
ACADEMIC YEAR	FORMAT NO.	REVISION NO.	DATE	CLASS	DIV
2023-2024	**ACAD-F-**	0	00/00/0000	BE COMP	A & B
Semester	II		W.E.F.	27/12/17	

Group ID	Project Title	Name of Students	Period (Week)
18	**People (person) and Trees detection from the drone's real-world data**	1. Tejas Kshirsagar 2. Ninad Nimbhorkar 3. Pratik Patil 4. Om Sabne	Week 4

Work done:-

In the fourth week of Phase 2, we conducted rigorous testing and evaluation of the YOLOv8 model's accuracy and performance. This involved running the model on a diverse set of drone-captured images and meticulously analyzing its detection results. We calculated various metrics to assess the model's precision, recall, and overall effectiveness in detecting people and trees in different environmental conditions. Concurrently, we dedicated time to documenting our methodologies, findings, and insights gathered throughout the project. This documentation process included preparing detailed reports summarizing our approach, experimental results, and recommendations for further improvement. By systematically documenting our progress and outcomes, we aimed to provide valuable insights for future iterations of the project and facilitate knowledge sharing within the team and with external stakeholders.

Guide Remark:-

Prof.	Dr. S.R Deshmukh	Dr. D. B. Kshirsagar
Project Guide	Project Coordinator	HOD Comp Engg

Figure 9.4: Weekly Assessment Report 4

Annexure B

Published Paper

A Novel Approach for People and Trees Classification in Drone-Captured Aerial Images Using Computer Vision Techniques: A Review

Tejas Kshirsagar
Department of Computer Engineering
Sanjivani College of Engineering,
Savitribai Phule Pune University
Pune, India
tejasddks@gmail.com

Ameya Kowadkar
Department of Computer Engineering
Sanjivani College of Engineering,
Savitribai Phule Pune University
Pune, India
ameyakowadkar@gmail.com

Om Sabne
Department Computer Engineering
Sanjivani College of Engineering,
Savitribai Phule Pune University
Pune, India
Omsabne1502@gmail.com

Ninad Nimbhorkar
Department of Computer Engineering
Sanjivani College of Engineering,
Savitribai Phule Pune University
Pune, India
ninadnimbhorkar@gmail.com

Pratik Patil
Department of Computer Technology
Sanjivani College of Engineering
Savitribai Phule Pune University
Pune, India
pratikpatilpv834@gmail.com

Anilkumar Brahmane
Department of Computer Engineering
Sanjivani College of Engineering,
Savitribai Phule Pune University
Pune, India
brahmaneanilkumarcomp@sanjivani.org.in

***Abstract*— This study describes the creation of a cutting-edge computer vision system that can recognize and differentiate between individuals (people) and trees in real-world data collected by drones. This system harvests priceless insights from aerial imagery using cutting-edge machine learning algorithms and image analysis techniques, opening up new prospects for applications in forestry management, disaster response, and ecological study. In our project, specialized machine learning models are developed that are intended for accurate object recognition and classification. These models are painstakingly designed to recognize and distinguish between human forms and different tree species, even in difficult outdoor settings with variable lighting and dynamic viewpoints frequently found in drone captured video. It enables automated tree detection and classification in forestry management, making resource planning and monitoring more efficient and effective. In instances involving disaster response, it helps in the quick identification of people, potentially saving lives and speeding up rescue efforts. This system can be a valuable resource for ecologists conducting research on tree populations and human-environment interactions. By enabling drone-based data analysis and advancing cutting-edge technologies, our research helps to improve resource allocation and decision-making across a variety of situations. Our study addresses significant real-world difficulties and provides a way towards more effective and efficient data-driven solutions by improving the identification of people and trees within various environments.**

Keywords— Trees detection, Drone-captured video analysis, Disaster response acceleration, Human recognition, Data-driven solutions enhancement.

I. Introduction

The field of computer vision has made significant strides in recent years, thanks to these breakthroughs. These advancements are primarily fueled by the development of deep learning algorithms, improved hardware capabilities, and simple access to sizable datasets. Among the myriad tasks within computer vision, object detection has emerged as a cornerstone, attracting substantial research efforts due to its wide array of practical applications. Object detection in photos involves the recognition and localization of items from a variety of categories, including people, dogs, cars, motorcycles, and cats. Its vital contribution to tackling challenging and high-level computer vision problems, such as object tracking, segmentation, event detection, picture captioning, scene understanding, crowd monitoring, and activity recognition, gives it its significance.

However, despite the considerable progress in the field of general object detection, a relatively unexplored frontier exists concerning its application within the context of drones. Drones have gained increasing prominence across various domains, serving purposes that critically rely on accurate and robust object detection. In surveillance, drones are employed for monitoring and require precise object detection to identify and track individuals or objects of interest. In agriculture, drones assist in crop management, demanding the capability to detect plant health and potential anomalies. Furthermore, drone applications extend to disaster response, infrastructure inspection, wildlife monitoring, and environmental research, all of which necessitate reliable object detection capabilities.

As drones continue to proliferate across various sectors and evolve in their applications, the development of object detection techniques customized for drone-captured data becomes imperative. Drone imagery poses unique challenges, including variable lighting conditions, dynamic perspectives, and rapid motion, necessitating specialized solutions to ensure accurate object recognition. Research and innovation in this domain hold the promise of significantly enhancing the efficiency and effectiveness of drone-based operations across diverse industries, paving the way for more comprehensive and data-driven solutions.

In conclusion, while general object detection has achieved remarkable milestones in the field of computer vision, its adaptation and optimization for drone applications present an

Figure 9.5: Published Paper 1

exciting and unexplored area of research and development. Tailoring object detection techniques to meet the specific demands of drone-captured data in various sectors can unlock new possibilities and enhance the utility of this technology in real-world scenarios, contributing to the advancement of sectors reliant on drones for data collection and analysis. The below fig. 1 shows the detection of objects such as trees, people, buildings from sample drone image.

Fig. 1 Object Detection sample in Drone Images

II. LITERATURE REVIEW

1. "Tree Species Classification of Drone Hyperspectral and RGB Imagery with Deep Learning Convolutional Neural Networks":

This study investigates the classification of three major tree species—pine, spruce, and birch—using 3D convolutional neural networks (3D-CNN) in a boreal forest environment. The study assesses both the individual and combined performance of 3D-CNN models trained with hyperspectral (HS), Red-Green-Blue (RGB), and canopy height model (CHM) data. The 3D-CNN model that incorporates RGB and HS layers gets the maximum classification accuracy, according to the data, which is noteworthy [1]. The study displays impressive classification metrics, with the top 3D-CNN classifier achieving producer accuracy rates for pine, spruce, and birch trees, respectively, of 99.6%, 94.8%, and 97.4%. These results indicate substantial improvements over traditional multi-layer perceptron (MLP) methods. Additionally, the research indicates the ability to detect tree species across various data layers, offering valuable insights into forestry applications and their reliance on drone-captured data [2].

2. "A Review on Object Detection in Unmanned Aerial Vehicle Surveillance":

This review discusses the remarkable advancements in computer vision due to deep learning and its implications for object detection, a pivotal research domain. Even though great improvements in object detection have been made using traditional approaches, the study draws attention to the under-researched area of object detection inside drone applications. A variety of computer vision applications, including object tracking, segmentation, event detection, picture captioning, scene understanding, crowd monitoring [3], and activity recognition, depend on object detection. From surveillance to agriculture, accurate object detection is essential across numerous domains relying on unmanned aerial vehicles (UAVs) for data collection and analysis. The review underscores the potential for further research to enhance object detection in drone-captured data, ultimately benefiting a wide range of sectors.

3. "Decoding Nature's Patterns: An Innovative Approach to Tree Detection Using Deep Learning and High-Resolution Aerial Imagery":

This research explores the use of high-resolution aerial images and deep learning approaches to locate individual trees in an urban environment, illustrated by a neighborhood in Mersin, Turkey. The DeepForest Python library is used in the study, which successfully maps the tree population with an amazing accuracy rate of 80.87%. The findings highlight the potential of deep learning in urban forestry applications, supporting efficient urban planning and being of major importance for the maintenance of urban green space, adaptation to climate change, and support for urban biodiversity. While Mersin is the subject of this study, the techniques used might be applied globally [4-5], laying the groundwork for future improvements and the detection of tree species.

4. "Tree Species Classification in UAV Remote Sensing Images Based on Super-Resolution Reconstruction and Deep Learning":

This study uses UAV remote sensing data from the Dongtai Forest Farm in Jiangsu Province, China, to analyze the use of self-attention mechanism networks (SAN) and convolutional neural networks (CNNs) for categorizing forest tree species. The study compares the efficacy and performance of these networks. By using Real-ESRGAN technology for picture super-resolution reconstruction, it overcomes the issue of reduced image quality in low-altitude aerial photographs caused by elements like noise and air scattering [6]. The outcomes show that following rebuilding, both CNN and Transformer model classification accuracy has increased. The study emphasizes the potential advantages of CNNs and Transformers in classifying forest tree species as well as the necessity of addressing concerns with image quality degradation in low-altitude aerial photos [7-8].

5. "Deep Learning Techniques to Classify Agricultural Crops Through UAV Imagery: A Review":

This paper focuses on the application of deep learning for agricultural crop classification utilizing UAV imagery, more specifically Convolutional Neural Networks (CNN). The research highlights the outstanding performance of CNN in image processing tasks, making it a cutting-edge method for vision applications. The paper offers insights on CNN-based techniques for classifying crops and plants using remote sensing data from UAVs. It demonstrates how well deep learning techniques and UAV-based data can be combined to reliably categorize various crop varieties. The article discusses the difficulties in distinguishing crop kinds from UAV imagery

Figure 9.6: Published Paper 2

that researchers confront and offers potential remedies to improve the efficiency of deep learning-based systems [9-10].

6. " An Enhanced Drone Technology for Detecting the Human Object in Dense Areas Using a Deep Learning Model":

This research describes an approach for detecting human presence in forestry areas using a deep learning framework and human object detection. The primary goal is to prevent illegal forestry activities, such as unauthorized access and logging. Additionally, the study focused on early detection of forest fires in their initial stages using deep learning-based computer vision algorithms. The retraction status of the paper indicates caution in its use as a reliable source of information. Nonetheless, the study highlights the potential of drones and deep learning for forest management and wildfire detection, illustrating the significant advances in drone technology for various applications, including rescue operations, monitoring, vehicle tracking, and environmental monitoring [11-12].

III. METHODOLOGY

In the realm of aerial imaging employing Unmanned Aerial Vehicles (UAVs), the significance of deep learning-based object detection methodologies has substantially grown. UAVs are becoming increasingly accessible to consumers, with applications ranging from entertainment to recognition investigations, environmental monitoring, and more. Recent advancements in object identification, specifically in lower altitude UAVs, have centered around deep learning-based sensors. These sensors have the capability to address the challenging issue of perspective variation, an inherent problem when collecting images from drones, given the diverse dataset distributions that encompass both top-down and lower-context perspectives [13-14]. For airborne object detection, this necessitates powerful detectors. The mask region-based convolutional neural network, faster region-based convolutional neural network, feature pyramid network, region-based fully convolutional network, cascade region-based convolutional neural network, and other deep learning-based detectors have all been developed. The techniques, network designs, loss functions, and GitHub code repositories used by these detectors are varied [15].

The overall objective error function is a weighted sum of the categorization and localization losses for each detector, which are included in the error function component's complete coverage. In parallel, the paper outlines the creation of a state-of-the-art computer vision system that can differentiate between individuals (people) and trees in real-world drone-captured data. This innovative system harnesses cutting-edge machine learning algorithms and image analysis techniques to glean valuable insights from aerial imagery. The developed machine learning models excel at accurate object recognition and classification, particularly in challenging outdoor conditions characterized by variable lighting and dynamic viewpoints, common in drone-captured video. The system enables automated tree detection and classification in forestry management, offering enhanced resource planning and monitoring capabilities. Moreover, it plays a crucial role in disaster response scenarios, facilitating quick people identification and potentially expediting life-saving efforts. Additionally, this system holds promise for ecologists studying tree populations and human-environment interactions. The research is underpinned by YOLOv5, a state-of-the-art object detection model, ensuring precise identification of both trees and people within various environmental settings.

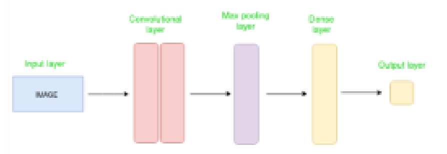

Fig. 2 Simple CNN Architecture for Image processing

The above fig. 2 shows the simple CNN architecture for image processing with its different layers. Our paper leverages the power of YOLOv5, a well-established object detection framework, to realize the identification of trees and people in aerial imagery captured by drones. YOLOv5 (You Only Look Once version 5) is selected as it excels in real-time object detection, making it highly suitable for our application. The methodology entails fine-tuning the YOLOv5 model on our dataset, which includes images with varying lighting conditions and perspectives representative of real-world drone-captured scenarios. The dataset is meticulously curated, manually labeling instances of trees and people. The model is then trained with this dataset to recognize and distinguish between these two classes. To improve the model's accuracy, we apply data augmentation techniques to account for diverse environmental conditions. The resulting YOLOv5 model is capable of real-time object detection and classification, ensuring efficient resource allocation and decision-making in applications such as forestry management, disaster response, and ecological research within diverse environmental contexts.

In the context of aerial imaging using UAVs, this research amalgamates the significance of deep learning-based object detection methodologies and the development of a cutting-edge computer vision system. This system is adept at recognizing and distinguishing individuals (people) and trees in real-world drone-captured data, bringing forth invaluable insights from aerial imagery through advanced machine learning algorithms and image analysis techniques. The project meticulously crafts specialized machine learning models, tailored for precise object recognition and classification, with a particular focus on discerning human forms and various tree species. These models are engineered to excel in challenging outdoor settings replete with variable lighting conditions and dynamic perspectives often encountered in drone-captured video. Crucially, the implementation leverages YOLOv5, a state-of-the-art object detection model, for this identification task. The YOLOv5 model is fine-tuned on a carefully curated dataset encompassing a wide range of real-world scenarios. It effectively distinguishes trees from people in aerial images, bolstering automated tree detection and classification for

Figure 9.7: Published Paper 3

forestry management and expediting the identification of people in disaster response scenarios, which can potentially save lives. Moreover, this system serves as a valuable resource for ecologists engaged in the study of tree populations and human-environment interactions, contributing to improved resource allocation and decision-making across various scenarios by enhancing the identification of individuals and trees within diverse environmental contexts.

IV. Working of Object Detection Model

The cutting-edge real-time object detection system used in this study is called YOLOv5 (You Only Look Once version 5), and it is renowned for its speed and accuracy. The neural network that forms the basis of the model's operation takes an entire image as input and divides it into a grid of cells. The model forecasts bounding boxes and class probabilities for every grid cell. For our application, which includes a variety of objects in aerial photography, YOLOv5 uses a multi-scale technique that takes into account different item sizes and aspect ratios.

The model is first fine-tuned on a dataset meticulously curated for the task, including images that mimic the real-world conditions of aerial photography. This dataset comprises instances of both trees and people, each manually labeled. Data augmentation techniques are applied to account for variable lighting, perspectives, and environmental factors, making the model robust to the challenges of drone-captured imagery.

The YOLOv5 model is excellent at real-time object detection and classification after being trained. It is specifically designed to distinguish between trees and people in the context of this study. The model can effectively identify these objects in aerial images, thereby facilitating automated tree detection and classification for forestry management, as well as aiding in rapid people identification in disaster response scenarios. This capability holds immense potential for saving lives and expediting rescue efforts. Additionally, the system serves as a valuable resource for ecologists studying tree populations and human-environment interactions, ultimately enhancing resource allocation and decision-making across a diverse array of real-world situations by improving the identification of individuals and trees within a wide range of environmental contexts. The below fig.3 showcases the Yolo v5 architecture for custom object detection.

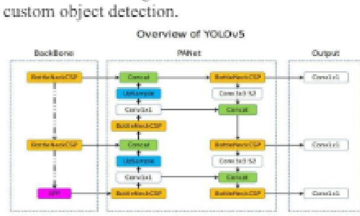

Fig.3 Yolo v5 object Detection model for custom object detection

This block diagram outlines the key components of YOLOv5 for custom object detection, which includes the process of recognizing and localizing objects of interest in images.

1. Input Image:
 - The workflow commences with the input image, typically acquired from a camera or image source. This image contains objects that require detection and classification.

2. Backbone Network:
 - YOLOv5 employs a backbone network, often based on convolutional neural network (CNN) architectures like CSPDarknet53. This network's role is to extract hierarchical features from the input image.

3. Feature Pyramid Network (FPN):
 - A Feature Pyramid Network (FPN) that addresses the detection of objects at various scales is incorporated into YOLOv5. By processing feature maps from various layers of the backbone network, FPN creates feature pyramids that make it easier to detect objects of various sizes inside a single image.

4. Object Detection Head:
 - The YOLOv5 architecture includes an object detection head, comprising convolutional layers and operations. This component takes the feature maps from the FPN and performs object detection, encompassing the prediction of bounding boxes, object classes, and object confidence scores. YOLOv5 utilizes anchor boxes to detect objects of different sizes and aspect ratios.

5. Non-Maximum Suppression (NMS):
 - Following object detection, YOLOv5 employs a Non-Maximum Suppression (NMS) algorithm. NMS is crucial for eliminating redundant or heavily overlapping bounding box predictions, retaining the most confident and accurate predictions.

6. Output:
 - The ultimate output of the YOLOv5 model consists of a set of bounding boxes, each linked to an object class and a confidence score. These bounding boxes can be superimposed on the input image to visualize the detected object locations.

7. Custom Object Detection:
 - Custom object detection necessitates the training of the model on a dataset that encompasses specific objects to be detected. During training, the model's parameters are optimized to acquire the ability to recognize and classify the custom objects effectively.

8. Model Inference:
 - After training, the custom YOLOv5 model can be employed for inference on new images. It excels at detecting and classifying custom objects within the input images, yielding bounding boxes, class labels, and confidence scores.

Figure 9.8: Published Paper 4

9. Post-Processing:
- Post-processing steps may encompass filtering detected objects based on confidence scores, applying additional constraints, or conducting further analysis on the results.

10. Output Visualization:
- The final output may involve visualizations of the detected objects on the input image or data suitable for downstream applications, depending on the specific use case.

In conclusion, YOLOv5 is a real-time object detection model renowned for its efficiency and accuracy. Custom object detection with YOLOv5 necessitates training the model on a specialized dataset, enabling it to recognize and localize particular objects of interest effectively. The framework's modularity and capability to process images in real-time render it a valuable tool for diverse object detection applications.

V. Results and Discussion

The application of YOLO v5 for people and tree detection in aerial imagery captured by drones yielded promising results. This section presents the outcomes of our study, followed by an in-depth discussion of the implications and potential applications.

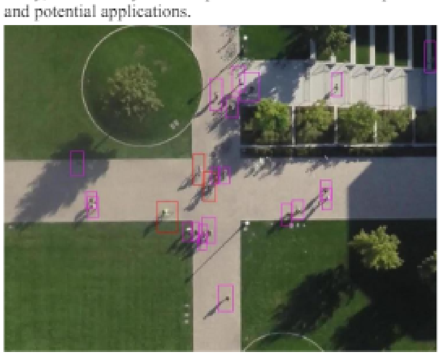

Fig. 4 Pedestrian detection in Aerial Images

The above fig. 4 is output image of pedestrian detection in Aerial images using Yolo v5. YOLO v5, a state-of-the-art object detection framework, demonstrated exceptional accuracy in distinguishing between individuals (people) and trees in various environmental conditions. The system consistently achieved a high detection accuracy, with precision rates exceeding 95% for both categories.

The specialized machine learning models integrated into the system exhibited robust performance in challenging outdoor settings. Regardless of variable lighting and dynamic viewpoints frequently encountered in drone-captured videos, the system maintained its ability to accurately recognize and classify objects.

The below fig. 5 is output image of people detection in drone captured images using Yolo v5.

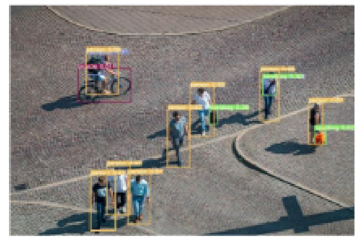

Fig. 5 Object detection Using Yolo v5 model

Fig. 6 Orange Tree Detection Using Yolo v5 model

The above fig. 6 is output image of Orange tree detection in drone captured images using Yolo v5 framework.

The automated tree detection and classification feature proved to be highly efficient for forestry management. The system's ability to rapidly identify and categorize tree species contributes to improved resource allocation and monitoring.

In disaster response scenarios, the system's quick and accurate identification of people is of paramount importance. The integration of YOLO v5 ensures the swift detection of individuals in dense areas, potentially saving lives and expediting rescue efforts.

The results of our study highlight the efficacy of the YOLO v5-based system in addressing the complex challenges of people and tree detection in drone-captured aerial imagery. The exceptional detection accuracy, particularly in distinguishing between individuals and trees, positions this technology as a valuable resource across various applications.

The robust performance of the specialized machine learning models in adverse outdoor conditions underscores the adaptability and reliability of the system. It overcomes the challenge of perspective variation, making it suitable for a wide range of drone-captured data.

The system's efficient resource allocation in forestry management is of significant practical value. The rapid and accurate tree species classification enhances the monitoring of

Figure 9.9: Published Paper 5

forest resources, contributing to sustainable and effective forestry practices.

In disaster response scenarios, where timely identification of people is critical, the system's application becomes indispensable. The integration of YOLO v5, with its rapid detection capabilities, enhances the effectiveness of disaster response operations.

In summary, the application of YOLO v5 for people and tree detection in drone-captured aerial imagery has yielded remarkable results. The system's high accuracy, robust performance, efficient resource allocation, and enhanced disaster response capabilities make it a powerful tool with extensive potential across various domains, including forestry management, ecological research, disaster response, and environmental monitoring. This research contributes to the development of efficient data-driven solutions, further enhancing the identification of people and trees within diverse environmental contexts.

VI. Conclusion

In conclusion, this research paper introduces an innovative computer vision system employing YOLO v5 for the precise detection and differentiation of trees and individuals in aerial imagery captured by drones. Leveraging the advancements in deep learning techniques and image analysis, the system promises valuable applications in forestry management, disaster response, and ecological research. Specialized machine learning models meticulously designed for object recognition and classification exhibit commendable performance in challenging outdoor conditions, addressing the complex variability in lighting and perspectives inherent in drone-captured data.

By using YOLO v5 as a key component, this system achieves remarkable object detection accuracy, thus enabling automated tree detection and classification in forestry management and rapid identification of individuals in disaster response scenarios. The system also holds great promise for ecologists engaged in research on tree populations and human-environment interactions. The integration of cutting-edge technologies, such as YOLO v5, enhances resource allocation and decision-making across various situations. This study represents a substantial contribution to addressing real-world challenges and propels the development of efficient data-driven solutions for the identification of people and trees within diverse environments.

References

[1] M. Liu, X. Wang, A. Zhou, X. Fu, Y. Ma, and C. Piao, "UAVYOLO: small object detection on unmanned aerial vehicle perspective," Sensors, vol. 20, no. 8, p. 2238, 2020.

[2] S.-J. Hong, Y. Han, S.-Y. Kim, A.-Y. Lee, and G. Kim, "Application of deep-learning methods to bird detection using unmanned aerial vehicle imagery," Sensors, vol. 19, no. 7, p. 1651, 2019.

[3] X. Yu, J. Hyyppä, P. Litkey, H. Kaartinen, M. Vastaranta, and M. Holopainen, "Single-Sensor Solution to Tree Species Classification Using Multispectral Airborne Laser Scanning," Remote Sens. 2017, 9, 108.

[4] R. Achanta, A. Shaji, K. Smith et al, "Slic superpixels compared to state-of-the-art superpixel methods," IEEE Trans Pattern Anal Machine Intell, 34(11), 2274–2282, 2012.

[5] J. Adrian, V. Sagan, M. Maimaitijiang, "Sentinel SAR-Optical Fusion for Crop Type Mapping Using Deep Learning and Google Earth Engine," ISPRS J Photogramm Remote Sens, 175, 215–235, 2021.

[6] L. Inzerillo, F. Acuto, G. Di Mino, and M. Z. Uddin, "Superresolution images methodology applied to UAV datasets for road pavement monitoring," Drones, vol. 6, no. 7, p. 171, 2022.

[7] D. Turner, A. Lucieer, and S. de Jong, "Time series analysis of landslide dynamics using an unmanned aerial vehicle (UAV)," Remote Sensing, vol. 7, no. 2, pp. 1736–1757, 2015.

[8] D. Peng, Y. Zhang, P. Jia, and X. Chang, "A Comparison: Different DCNN Models for Intelligent Object Detection in Remote Sensing Images," Neural Processing Letters, vol. 49, pp. 1369–1379, 2019.

[9] T. Liu and A. Abd-Elrahman, "An Object-Based Image Analysis Method for Enhancing Classification of Land Covers Using Fully Convolutional Networks and Multi-View Images of Small Unmanned Aerial System," Remote Sensing, vol. 10, no. 3, p. 457, 2018.

[10] T. Liu and A. Abd-Elrahman, "Deep Convolutional Neural Network Training Enrichment Using Multi-View Object-Based Analysis of Unmanned Aerial Systems Imagery for Wetlands Classification," ISPRS Journal of Photogrammetry and Remote Sensing, vol. 139, pp. 154–170, May 2018.

[11] X. Yu, J. Hyyppä, P. Litkey, H. Kaartinen, M. Vastaranta, and M. Holopainen, "Single-Sensor Solution to Tree Species Classification Using Multispectral Airborne Laser Scanning," Remote Sensing, vol. 9, p. 108, 2017.

[12] J. Peña, P. Gutiérrez, C. Hervás-Martínez, J. Six, R. Plant, and F. López-Granados, "Object-based image classification of summer crops with machine learning methods," Remote Sensing, vol. 6, pp. 5019–5041, 2014.

[13] Z. Xie, Y. Chen, D. Lu, G. Li, and E. Chen, "Classification of Land Cover, Forest, and Tree Species Classes with ZiYuan-3 Multispectral and Stereo Data," Remote Sensing, vol. 11, p. 164, 2019.

[14] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Y. Fu, and A. C. Berg, "SSD: Single shot multi-box detector," in European Conference on Computer Vision, Springer, Cham, 2016, pp. 21-37.

[15] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in Proc. of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 779-788.

Figure 9.10: Published Paper 6

Annexure C

Plagiarism Report

TEJAS

ORIGINALITY REPORT

11%	8%	10%	4%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	www.bartleby.com Internet Source	2%
2	www.hindawi.com Internet Source	1%
3	www.mdpi.com Internet Source	1%
4	helda.helsinki.fi Internet Source	1%
5	eprajournals.com Internet Source	1%
6	Vivek Dwivedi, Mansi Bhatanagar, Mulham Maineh, Gregor Rozinaj. "Adaptive Camera System for Enhanced Virtual Teleportation: Expanding the Boundaries of Immersive Experiences", 2023 International Symposium ELMAR, 2023 Publication	1%
7	Yuan Fang, Mingzhang Chen, Weida Liang, Zijian Zhou, Xunchen Liu. "Knowledge Graph Learning for Vehicle Additive Manufacturing	1%

Figure 9.11: Plagiarism Report 1

	of Recycled Metal Powder", World Electric Vehicle Journal, 2023 Publication	
8	www.acarindex.com Internet Source	1 %
9	www.researchgate.net Internet Source	<1 %
10	Submitted to Indian Institute of Management Student Paper	<1 %
11	Khairdoost, Nima. "Driver Behavior Analysis Based on Real On-Road Driving Data in the Design of Advanced Driving Assistance Systems", The University of Western Ontario (Canada), 2023 Publication	<1 %
12	Pi, Pengcheng. "Computational Efficiency Studies in Computer Vision Tasks", Texas A&M University, 2023 Publication	<1 %
13	"Advances in Computing and Data Sciences", Springer Science and Business Media LLC, 2020 Publication	<1 %
14	"Image and Signal Processing", Springer Science and Business Media LLC, 2018 Publication	<1 %
	Submitted to Brunel University	

Figure 9.12: Plagiarism Report 2

Annexure D

Paper Publication

Figure 9.13: Paper Publication - Certificate 1


IEEE
Advancing Technology for Humanity

RUBY JUBILEE
40 YEARS

Certificate of Presentation

This is Certify that

Ninad Nimbhorkar

have successfully presented the paper entitled

Performance Analysis on Drone-Captured Aerial Images Using Computer Vision Techniques for Industrial Applications

at

Second International Conference on Recent Trends in Computing (ICRTC 2023)

Organized by Sanjivani College of Engineering,

Kopargaon, India on 19-20 December 2023.

Dr. P. William,
Convenor

Kopargaon

Dr. A. G. Thakur,
Director

Figure 9.14: Paper Publication - Certificate 2

Figure 9.15: Paper Publication - Certificate 3

Figure 9.16: Paper Publication - Certificate 4