VEGETATION ENCROACHMENT RISK ASSESSMENT NEAR ELECTRIC POLES USING IMAGE PROCESSING

Project Report submitted by

SHELDON MATHIAS
(4NM20CS166)

SHRI HARI S
(4NM20CS177)

YASHASWI P
(4NM20CS216)

RAJATH
(4NM21CS413)

Under the Guidance of

Mrs. SHILPA KAREGOUDAR
Asst. Professor Gd-II,
Dept. of CSE,
NMAMIT

In partial fulfillment of the requirements for the award

of the Degree of

Bachelor of Engineering in Computer Science and Engineering

from

Visvesvaraya Technological University, Belagavi

Department of Computer Science and Engineering
NMAM Institute of Technology, Nitte - 574110
(An Autonomous Institution affiliated to VTU, Belagavi)

APRIL 2024

N.M.A.M. INSTITUTE OF TECHNOLOGY
(An Autonomous Institution affiliated to Visvesvaraya Technological University, Belagavi)
Nitte – 574 110, Karnataka, India

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

Certified that the project work entitled

"Vegetation Encroachment Risk Assessment near Electric Poles using Image Processing"

is a bonafide work carried out by

Sheldon Mathias(4NM20CS166) ***Shri Hari S(4NM20CS177)***

Yashaswi P(4NM20CS216) ***Rajath(4NM21CS413)***

in partial fulfilment of the requirements for the award of

Bachelor of Engineering Degree in Computer Science and Engineering

prescribed by ***Visvesvaraya Technological University, Belagavi***

during the year ***2023-2024****.*

It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the report deposited in the departmental library.

The project report has been approved as it satisfies the academic requirements in respect of the project work prescribed for the Bachelor of Engineering Degree.

Signature of the Guide	Signature of the HOD	Signature of the Principal

Semester End Viva Voce Examination

Name of the Examiners	Signature with Date
1.	
2.	

ACKNOWLEDGEMENT

Any achievement, be it scholastic or otherwise does not depend solely on the individual efforts but on the guidance, encouragement and cooperation of intellectuals, elders, and friends. Several personalities, in their own capacities have helped me in carrying out this project work. We would like to take this opportunity to thank them all.

First and foremost, we would like to thank **Dr. Niranjan N Chiplunkar**, Principal, NMAMIT, Nitte, for his moral support towards completing our project work.

We would like to thank **Dr. Jyothi Shetty**, Head of the Department, Computer Science and Engineering, NMAMIT, Nitte, for her valuable suggestions and expert advice.

We deeply express our sincere gratitude to our guide **Mrs. Shilpa Karegoudar,** Department of CSE, NMAMIT, Nitte, for her able guidance, regular source of encouragement and assistance throughout this project work.

We thank our parents and all the faculty members of the Department of Computer Science and Engineering for their constant support and encouragement.

Last, but not the least, we would like to thank our peers and friends who provided us with valuable suggestions to improve our project.

Sheldon Mathias(4NM20CS166)
Shri Hari S(4NM20CS177)
Yashaswi P(4NM20CS216)
Rajath (4NM21CS413)

ABSTRACT

In contemporary urban landscapes, the coexistence of urban trees and power lines poses a multifaceted challenge characterized by the escalating complexity of managing entanglements. Rapid urbanization, coupled with the imperative for climate adaptation and environmental services, necessitates a comprehensive understanding of the intricate interplay between these vital elements. The overarching problem lies in the potential safety hazards, service disruptions, and ecological consequences arising from tree-power line entanglements, demanding innovative and automated solutions for effective detection and management. By using modern day techniques such as Image Processing and Deep learning, we plan to identify the regions that have vegetation-wire entanglement so that necessary action can be taken.

Keywords: Image Processing, Deep Learning, Vegetation Encroachment, Electric power lines

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1
INTRODUCTION

Electric power distribution systems play a vital role in supplying energy to communities, industries, and households. However, the uninterrupted flow of electricity is contingent upon the efficient operation and maintenance of the infrastructure, particularly the electric poles that form the backbone of the distribution network. One critical challenge faced in this regard is the encroachment of vegetation in the vicinity of these poles. Vegetation encroachment poses a significant threat to the reliability and safety of the power distribution system. Overgrown vegetation can lead to power outages, equipment damage, and even pose fire hazards. Traditional methods of manual inspection are not only labor-intensive but also time- consuming. To address these challenges, this report presents a novel approach leveraging image processing techniques for the automated assessment of vegetation encroachment risks near electric poles.

The objective of this project is to develop an efficient and accurate system that can analyze images captured in the field, identify the presence of vegetation, and assess the level of risk associated with its proximity to electric poles. Through the integration of advanced image processing algorithms, this approach aims to enhance the overall reliability of power distribution systems while reducing the need for extensive manual inspections. This report outlines the methodology employed, including data collection processes, image processing techniques, and risk assessment methodologies. The findings of this study provide valuable insights into the feasibility and effectiveness of utilizing image processing for vegetation management in the context of power distribution infrastructure.

As we delve into the details of our methodology and results, it becomes apparent that the integration of image processing technologies offers a promising avenue for proactive vegetation management, ensuring the resilience and sustainability of electric power distribution systems.

Electric power distribution systems are susceptible to disruptions caused by various external factors, and vegetation encroachment emerges as a recurring challenge with the potential for severe consequences. The increasing demand for electricity, coupled with changing climate patterns, has heightened the importance of robust infrastructure management. Traditional methods of vegetation monitoring often fall short in providing timely and comprehensive assessments. This project responds to the pressing need for a more proactive and technologically advanced approach to vegetation management near electric poles. By harnessing the capabilities of image processing, we aim to create a system that not only identifies the presence of vegetation but also evaluates the associated risk levels. This transition towards automation not only enhances the efficiency of the inspection process but also allows for more frequent and consistent monitoring.

The integration of image processing techniques promises a paradigm shift in how we perceive and address vegetation encroachment risks. The ability to analyze large volumes of image data rapidly empowers utility companies to make informed decisions in real-time, mitigating potential risks before they escalate. As we embark on this exploration of image processing applications, the intention is to contribute to the ongoing dialogue on innovative solutions for enhancing the resilience and reliability of power distribution networks. Through a detailed examination of our methodology, results, and subsequent discussions, this report endeavors to showcase the potential impact of image processing in the realm of vegetation encroachment risk assessment. By doing so, we not only advance the field of infrastructure management but also pave the way for sustainable and adaptive approaches to challenges that stand at the intersection of technology and utility services.

CHAPTER 2

LITERATURE SURVEY

[1] This study used UAV-borne LiDAR and the Richards growth model to detect tree encroachments in high voltage powerline corridors. UAV-borne LiDAR was effective in accurately measuring tree heights. The Richards model proved to be the best fit for modeling tree growth. A bounding box-based algorithm greatly improved detection efficiency, and a 50-meter unit length was recommended for segmenting powerline buffers. The study suggested targeted inspections to enhance efficiency but noted challenges in species recognition and environmental factors affecting powerline positions.

[2] The paper introduces a new method for monitoring power line corridors using satellite imagery, addressing the challenge of detecting transmission towers (TTs) and vegetation encroachment (VE) efficiently and cost-effectively. Traditional methods like UAVs and airborne photography are expensive and impractical for wide-area monitoring.

[3] A combination of LiDAR data with aerial and radar imagery allows to track dynamic seasonal growth of vegetation around critical infrastructure such as power lines. It present a general framework that integrates tree identification and growth assessment around power lines with the goal to identify locations of high risk where trees potentially cause power outages

[4] In this paper, a novel technique for depth estimation of vegetation/trees is proposed. In the study, Dynamic Programming is employed on stereo satellite images to determine depth of vegetation and trees. The experimental results on Quick Bird imagery exhibit that the proposed technique performs better compared to block matching technique in terms of accuracy.

[5] The TTPLA dataset is introduced for aerial object detection and segmentation, with a focus on transmission towers (TTs) and power lines (PLs). It addresses challenges like diverse views, background complexity, and crowded objects. The dataset is created from aerial videos captured by UAVs, and it follows an instance segmentation annotation policy for precise object recognition.

[6] A knowledge-based power line detection method for a vision-based UAV surveillance and inspection system. A PCNN filter is developed to remove background noise from the images prior to the Hough transform being employed to detect straight lines. Finally knowledge-based line clustering is applied to refine the detection results.

[7] The paper proposes an intelligent detection framework for monitoring vegetation encroachment on power lines, combining deep learning and advanced stereovision techniques. It aims to enhance grid stability by detecting potential circuit failures caused by vegetation growth. The framework consists of three modules: vegetation region detection using Faster R-CNN, power line detection via the Hough transform, and vegetation encroachment detection with an advanced stereovision algorithm. While promising, the framework lacks comprehensive evaluation, scalability considerations, and validation with real-world data, raising questions about its practical reliability.

[8] The excerpt discusses the significance of the "DALES Objects" dataset, a large-scale dataset for instance segmentation in aerial LiDAR data. It emphasizes the limited availability of instance segmentation datasets and highlights the dataset's scope, which includes semantic and instance segmentation labels, original point data in UTM projection, and intensity information. The dataset aims to support research in 3D deep learning for both LiDAR and outdoor scenes.

[9]In this paper, investigated the critical wind speed that heightens the risk of wildfires by calculating the distance between trees and wires. To conduct this study, we used airborne LiDAR data collected from Sonoma County in northern California and analyzed the behavior of a sample tree having a height of 19.2 m under wind loads. Our analysis showed that the main factor determining tree deflection is the ratio of the tree height to the trunk diameter.

[10] This paper discusses the need for research in extracting high voltage power lines from high-resolution satellite images. It introduces a new algorithm called Cluster Radon Transform (CRT) for this purpose, emphasizing its anti-noise capabilities and efficiency. The passage concludes by noting the potential for further research on extracting curved and broken power lines.

CHAPTER 3
PROBLEM DEFINITION

3.1 PROBLEM STATEMENT

In the dynamic urban landscape, the intricate interaction between urban trees and power lines gives rise to a significant problem. As cities expand, the challenge is to manage the entanglements between these elements, leading to safety concerns, service interruptions, and ecological repercussions. Recognizing the need for a nuanced understanding of these intersections, this research focuses on developing innovative solutions leveraging image processing and machine learning techniques. By exploring advanced technologies, the project aims to detect tree-power line entanglements efficiently. To address this pressing issue, this research initiative is dedicated to pioneering an innovative solution cantered around the application of image processing techniques for vegetation encroachment risk assessment near electric poles. By harnessing the capabilities of digital imagery, the primary objective is to automate the detection and analysis of vegetation growth in close proximity to electric infrastructure. This approach promises to empower utility companies and municipal authorities to efficiently identify areas of concern and prioritize maintenance efforts accordingly.

3.2 OBJECTIVES

Key objectives of the project include:

- Collect diverse satellite images encompassing both dense and non-dense vegetation to ensure comprehensive data acquisition for analysis
- Develop a Machine Learning Model
- Analyze and validate the model for existing data
- Integrate this model with Google Earth API such that the image of the region of interest can be directly extracted from Google Earth
- Map a power line in a region and identify locations where there is vegetation encroachment present near power lines model to classify each region of interest, and visualize the results by overlaying them on the satellite imagery

CHAPTER 4
SOFTWARE REQUIREMENTS AND SPECIFICATIONS

4.1 INTERFACE REQUIREMENTS

4.1.1 Software Requirements

Python 3: It offers a robust foundation due to its versatility and extensive libraries. Whether you're developing a web application, data analysis tool, or automation script, Python's clear syntax and rich ecosystem empower you to build efficiently and effectively. Leveraging frameworks like Django or Flask for web development, pandas for data manipulation, or TensorFlow for machine learning, Python 3 provides the tools necessary to bring your project to life. Additionally, its active community and wealth of online resources ensure you have the support needed to tackle any challenge. With Python 3, your project is poised for success.

HTML (Hypertext Markup Language): It is the standard markup language for creating web pages and web applications. It defines the structure of content on a web page using a system of tags and attributes. HTML documents are comprised of elements, which are represented by tags enclosed in angle brackets. These tags describe the purpose or meaning of the content they enclose, such as headings, paragraphs, images, links, and more. Attributes provide additional information about an element, such as its style or behavior. By combining various HTML elements and attributes, developers can create rich and interactive web experiences. Additionally, HTML works in conjunction with CSS (Cascading Style Sheets) and JavaScript to control the presentation and behavior of web pages, making it an essential skill for web development.

Visual Studio Code (VS Code): It is a popular, free source-code editor developed by Microsoft. It offers a wide range of features tailored for web and software development, including support for multiple programming languages, code debugging, version control integration, and an extensive library of extensions. VS Code's intuitive user interface, customizable layout, and powerful editing capabilities make it a favorite among developers of all skill levels. Its built-in terminal allows for seamless interaction with command-line tools,while its integration with Git facilitates collaborative development workflows. With its active community and continuous updates, Visual Studio Code remains a top choice for developers seeking a lightweight yet powerful coding environment.

4.1.2 Package requirements

- **NumPy:** For the Python programming language, NumPy is a library that adds support for sizable, multi-dimensional arrays and matrices. It also provides a sizable number of high-level mathematical functions to work with these arrays.
- **TensorFlow:** TensorFlow is a popular open-source framework for machine learning and deep learning. It provides a wide range of tools for developing and training models, including those for semantic segmentation of images. TensorFlow's powerful computational graph allows for efficient processing of large datasets, making it a popular choice for many semantic segmentation applications. The framework provides a wide range of pre-built models and tools for training custom models, making it an accessible choice for both beginners and experts in the field.
- **Matplotlib:** Matplotlib is a data visualization library for creating static, animated, and interactive visualizations in Python. It provides an extensive range of customizable visualizations, including line plots, scatter plots, bar plots, histograms, and many more. With its intuitive API, Matplotlib enables users to create high-quality and publication- ready plots and visualizations with ease.
- **Pandas:** Pandas is the cornerstone of data analysis and manipulation in Python. It introduces powerful data structures like Series (one-dimensional data) and DataFrames (tabular data, like spreadsheets). With pandas, you can efficiently load, clean, reshape, and analyze various datasets.
- **Sklearn (scikit-learn):** Sklearn is your go-to machine learning toolkit in Python. It provides a wealth of algorithms for tasks like classification (predicting categories), regression (predicting continuous values), clustering (grouping data points), dimensionality reduction (simplifying data), model selection, and preprocessing. sklearn offers a clean and consistent interface, making it incredibly user-friendly for building and evaluating machine learning models.

4.2 HARDWARE REQUIREMENTS:

- Operating System: Windows 11 (64-bit)
- Processor: Intel Core i5 or equivalent
- RAM: Minimum 8 GB
- Storage: Minimum 256 GB
- Graphics Card: Dedicated GPU with CUDA support

CHAPTER 5
SYSTEM DESIGN

Our system design orchestrates machine learning algorithms, image processing techniques, and feature extraction methods to analyze vegetation density near power lines using satellite imagery. By amalgamating these components, we create a robust framework capable of accurately classifying dense vegetation encroachment. This holistic approach ensures efficient preemptive maintenance strategies, thereby bolstering the resilience and security of power line infrastructure.

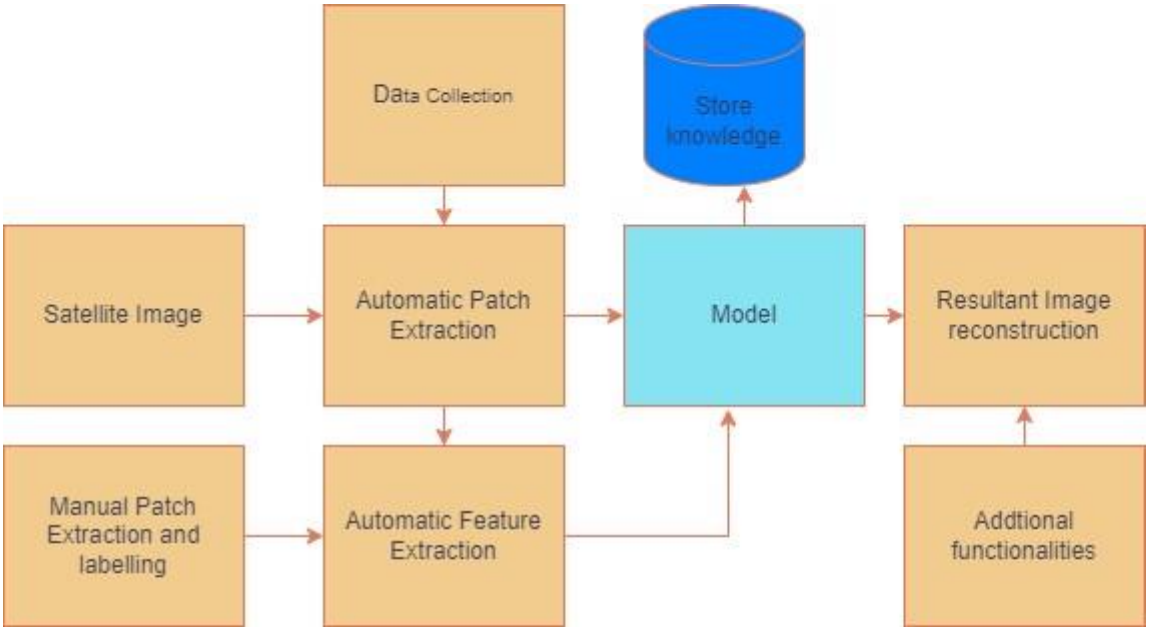

Fig 5.1 Proposed System Design

5.1 Procedural design

- **Data collection and preprocessing:**
  - **Data collection:** The initial phase of the project involves collecting images from Google Earth and processing them into smaller chunks of size 32x32. These chunks are manually classified as either Dense or Non-Dense vegetation, providing labeled data for subsequent analysis. This manual classification process ensures that the dataset accurately represents the diversity of vegetation types and densities present in the images. Once classified, the chunks are stored in the system, forming the basis of the dataset for further analysis and model development.

  - **Feature Extraction:** After dataset preparation, the next step is feature extraction, where various features of the image chunks are computed and stored in a structured format. These features include statistical measures such as Mean, Variance, and Standard Deviation, which provide insights into the characteristics of the vegetation present in each chunk. By extracting these features, the dataset is enriched with quantitative information that can be used to train machine learning models effectively. Texture features such as energy, ASM (Angular Second Moment), contrast, and homogeneity, which provide insights into the spatial arrangement of pixel intensities. By operating in different color spaces (BGR, HSV, LAB), the function ensures that a comprehensive range of features is captured, accounting for variations in color representation and perception.
  - **Data Splitting:** With the dataset prepared and features extracted, the data is then split into separate training and testing sets in the ratio of 80% for training and 20% for testing. This data splitting ensures that the models are trained on a sufficient amount of data while also allowing for independent evaluation to assess their performance accurately. By partitioning the data in this manner, the models can be trained on one subset and evaluated on another, enabling robust validation of their predictive capabilities.

- **Model Building:**
  - **Machine Learning model:** In this phase, various machine learning models are developed using the training data to classify vegetation chunks as Dense or Non-Dense. These models encompass a range of algorithms, including but not limited to Decision Trees, Random Forests, and Neural Networks. Each model is trained on the training dataset using the extracted features, aiming to learn patterns and relationships that distinguish between Dense and Non-Dense vegetation.
  - **Model Evaluation:** Once the models are trained, they are evaluated using the testing dataset to assess their performance and identify the most effective model for the task. Evaluation metrics such as accuracy, precision, recall, and F1-score are calculated to measure the models' ability to correctly classify vegetation chunks. Additionally, techniques like cross-validation may be employed to ensure the robustness of the evaluation process. The model that demonstrates the highest performance metrics on the testing dataset is selected for use in subsequent phases of the project, where it will be deployed for prediction and further analysis.

- **Visualization:**
  - **Image reconstruction and additional functionalities:** Following the prediction on new data using the trained DNN model, the system proceeds with the reconstruction of the resultant image, leveraging the predicted classifications for individual image chunks. Concurrently, it integrates additional functionalities, including a Satellite Imagery Downloader and utilities for distance calculation and pixel mapping. This integration enhances both data acquisition efficiency and analysis accuracy. Through the incorporation of these functionalities, the system achieves a seamless workflow, from data collection and preprocessing to model deployment and result visualization. The reconstructed image, highlighting areas of dense vegetation encroachment near power lines, is pivotal for providing actionable insights to decision-makers. By combining image reconstruction with these supplementary features, the system offers a comprehensive solution for vegetation density analysis and preemptive maintenance efforts in power line infrastructure management.

- **Image visualization:** To visualize the predicted areas of dense vegetation encroachment near power lines, we integra te the output of the trained DNN model with a visual representation of the satellite image. Leveraging the draw_square function, we mark areas identified as dense vegetation with squares overlaid onto the image. These squares serve as indicators of potential risk zones, aiding in preemptive maintenance planning. Additionally, we employ color-coding or other visual cues to distinguish between areas of dense vegetation and other regions, enhancing the interpretability of the results. By combining the model's predictions with visual indicators, decision-makers gain actionable insights into vegetation density patterns. This integrated approach facilitates efficient identification and prioritization of areas requiring attention, streamlining vegetation management efforts and ensuring the reliability and safety of power line infrastructure.

5.2 Algorithm Design:

- **Algorithm 1:Calculate nearest powerline segment**
  - Input Data:
    - Latitude and longitude value of the target area
  - Output Data:
    - ID of the nearest powerline segment from the database
  - Logic:
    - Sort the segments in terms of region and calculate haversine distance for each of the powerline segments in the region
    - Save the shortest segment ID and compare with the remaining segments by repeating the above steps

- **Algorithm 2:Latitude and longitude values to pixel mapping**
  - Input Data:
    - Image and latitude and longitude value of the top left corner and bottom right corner of the image with latitude and longitude value to be mapped inside the image along with the size of the image
  - Output Data:
    - Returns the pixel value of target location
  - Logic:
    - Calculate no of pixels in the diagonal of the image
    - Determine magnitude of the distance that each pixel occupies
    - Use the values to map to any other pixels inside the image

- **Algorithm 3:Encroachment detection**
  - Input Data:
    - Output from the model and target latitude and longitude value
  - Output Data:
    - State of the encroachment
  - Logic:
    - Draw a square of side 100 pixels taking target value as center
    - Check whether any pure red pixels are inside the square
    - If present return encroachment is detected in the area

CHAPTER 6

METHODOLOGY

- **Data Collection and Preprocessing:** High-resolution satellite images from Google Earth covering regions with power lines are segmented into 32x32 pixel chunks for localized analysis. Manual classification of these chunks as Dense or Non-Dense vegetation is conducted, laying the groundwork for feature extraction

- **Feature Extraction and Labeling:** Statistical features like Mean, Variance, and Standard Deviation are extracted from each chunk. Simultaneously, chunks are labeled based on vegetation density, creating a labeled dataset for model training

- **Data Splitting:** The labeled dataset is split into training and testing sets (80/20 ratio) to train the model on a significant portion and evaluate its generalization performance on a separate portion

- **Machine Learning Model Development:** A Deep Neural Network (DNN) model is constructed to classify vegetation density using extracted features. DNNs excel in learning intricate patterns, offering high accuracy in distinguishing dense and non-dense vegetation

- **Model Evaluation:** The DNN model's performance is assessed using metrics like accuracy, precision, and recall to guide the selection of the most effective model for deployment

- **Prediction on New Data:** Using the trained DNN model, predictions are made on new satellite image chunks to identify areas of dense vegetation encroachment near power lines, aiding in pre-emptive maintenance efforts

- **Result Image Reconstruction**: Predicted classifications for individual image chunks are arranged to recreate the complete image, facilitating visualization of vegetation density patterns

- **Visualization and Analysis:** The reconstructed image highlights areas of dense vegetation encroachment near power lines using color-coded visual indicators, providing actionable insights for decision-makers

- **Additional Functionalities:**Integrated functionalities include a Satellite Imagery Downloader and utilities for distance calculation and pixel mapping, enhancing data acquisition efficiency and analysis accuracy

CHAPTER 7
IMPLEMENTATION

7.1 Dataset Gathering:

The initial phase of the project involves collecting images from Google Earth and processing them into smaller chunks of size 32x32. These chunks are manually classified as either Dense or Non-Dense vegetation, providing labeled data for subsequent analysis. This manual classification process ensures that the dataset accurately represents the diversity of vegetation types and densities present in the images. Once classified, the chunks are stored in the system, forming the basis of the dataset for further analysis and model development.

	mean_r	mean_g	mean_b	mean_h	mean_s	mean_v	mean_l	mean_a	mean_bhuman
0	55.852539	64.175781	40.416016	40.208008	97.095703	64.208984	65.666992	119.663086	141.447266
1	46.280273	64.695312	38.921875	51.441406	109.980469	64.695312	63.858398	114.857422	141.323242
2	47.495117	62.459961	38.364258	47.905273	103.061523	62.509766	61.967773	116.664062	140.768555
3	41.094727	91.336914	80.406250	83.426758	142.566406	91.336914	89.491211	108.134766	129.188477
4	23.763672	54.360352	62.622070	95.867188	168.500977	62.731445	52.767578	119.776367	119.601562

Fig 7.1 Sample from Dataset

7.2 Feature Extraction:

Feature extraction is a fundamental step in image processing and computer vision tasks. In this section of the code, the extract_features function is defined to extract a diverse set of features from input images. These features encompass statistical properties such as mean, standard deviation, and variance, which capture essential aspects of the image's color distribution. Moreover, the function computes texture features such as energy, ASM (Angular Second Moment), contrast, and homogeneity, which provide insights into the spatial arrangement of pixel intensities. By operating in different color spaces (BGR, HSV, LAB), the function ensures that a comprehensive range of features is captured, accounting for variations in color representation and perception.

7.3 Model Defining:

In deep learning, model defining involves specifying the architecture and parameters of the neural network. This includes choosing the number of layers, types of activation functions, and connections between neurons. It's a crucial step as it determines the model's capacity to learn and represent complex patterns in data. As we are using Keras for the purpose of designing and implementing the network, the model is defined by adding various layers used and we compile the model before training as shown in the figure 4.

```
# Create a DNN model
model = Sequential()
model.add(Dense(128, activation='relu', input_shape=(X_train.shape[1],)))
model.add(Dropout(0.2))
model.add(Dense(64, activation='relu'))
model.add(Dropout(0.2))
model.add(Dense(1, activation='sigmoid'))

# Compile the model
model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])

# Train the model
history = model.fit(X_train, y_train, epochs=100, batch_size=32,validation_data=(X_val,y_val))
```

7.2 Model Definition

7.4 Model Training:

Model training is the process of feeding data into the defined neural network to adjust its parameters through backpropagation and optimization algorithms like gradient descent. During training, the model learns to map inputs to outputs by minimizing the difference between predicted and actual values, a process often iterated over multiple epochs to improve performance. The way it is done by using the inbuilt fit function defined in the Keras library by passing the training data to train the model.

7.5 Model Testing:

Model testing involves evaluating the trained model's performance on unseen data to assess its generalization ability. This step helps determine if the model can effectively make accurate predictions on real-world data beyond the training set. The metrics used for evaluation are accuracy score and loss in this model.

7.6 Model Evaluation:

Model saving entails persisting the trained neural network's parameters and architecture to disk for later use. This is crucial for deploying the model in production environments or sharing it with others for further analysis. This is done by using the save function in the keras library that takes in the filename and saves the

Accuracy: Accuracy simply measures how often the classifier correctly predicts. We can define accuracy as the ratio of the number of correct predictions and the total number of predictions.

$$\textit{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

Precision: Precision reveals how many of the situations that were predicted with accuracy ended up being positive. When false positives are more problematic than false negatives, precision is helpful. In e-commerce websites, music or video recommendation systems, and

$$Precision = \frac{TruePositive}{TruePositive + FalsePositive}$$

other applications where inaccurate results could drive away customers and hurt a business, accuracy is essential.

Recall (Sensitivity): How many of the actual positive cases our model was able to accurately forecast is known as recall. It is a helpful indicator when a False Negative is more problematic than False Positive.

$$Recall = \frac{TruePositive}{TruePositive + FalseNegative}$$

F1 Score: It gives a combined idea about Precision and Recall metrics. It is maximum when Precision is equal to Recall. F1 Score is the harmonic mean of precision and recall.

$$F1 = 2.\,\frac{Precision \times Recall}{Precision + Recall}$$

7.7 Other functionalities and & GUI implementation:

The envisioned application for geospatial analysis and risk detection integrates a suite of functionalities designed to handle various aspects of image processing, geographic data management, and analytical tasks. At its core, the Image Processing Module empowers users to manipulate images efficiently, allowing for tasks such as loading images from diverse sources, defining regions of interest within them, extracting relevant features, and predicting colors or classes associated with these features. This module, bolstered by libraries like PIL and numpy, serves as the foundation for subsequent analysis. Complementing this, the Google Maps Integration Module facilitates seamless interaction with the Google Maps API, enabling tasks such as downloading map tiles, stitching them into comprehensive map images, and projecting geographic coordinates onto the 2D map plane. This integration provides users with a powerful tool for visualizing and analyzing geographic data. The Database Management Module enhances the application's capabilities by providing robust data storage, retrieval, and manipulation functionalities. It enables users to add, sort, search, and export geographical data efficiently, ensuring data integrity and accessibility. Meanwhile, the Geospatial Analysis Module offers advanced analytical tools for processing geographical data and deriving meaningful insights. From calculating distances between coordinates to identifying nearest features and mapping coordinates onto images or maps, this module enables users to perform sophisticated spatial analyses. Lastly, the Risk Detection Module focuses on identifying risks, anomalies, or patterns within images or geographic data. By drawing regions of interest, counting pixels, applying machine learning algorithms, and generating alerts or reports, this module assists users in detecting and mitigating potential hazards effectively. By seamlessly integrating these functionalities and providing a user-friendly interface, the application offers a comprehensive solution for addressing complex geographical challenges across various domains.

CHAPTER 8
RESULTS

The data split was done in the ratio 60 – 20 – 20 to training validation and testing purpose. The training dataset contains 7410 samples of data stored in 232 batches. The testing and validation dataset contains 2471 samples of data stored in 78 batches The model is trained for 100 epochs. The train time for each epoch was about 1s with 3ms for each step.

During the initial stages of training, both the training and validation losses are observed to be high, while the accuracy remains relatively low. However, as training progresses there is a noticeable decrease in the loss metrics alongside an increase in accuracy for both the training and validation sets. This trend signifies that the model is effectively learning from the provided data, enhancing its performance over successive epochs. As the training nears completion, the loss values stabilize, and the accuracy metrics reach a plateau. This stabilization indicates that the model may have converged to a certain extent, suggesting that further training iterations might not significantly improve its performance. Overall, the observed patterns in loss reduction and accuracy improvement throughout the training process reflect the model's capability to learn from the data and optimize its predictive capabilities, ultimately contributing to its overall effectiveness in making accurate predictions.

The plotted figures depict the performance metrics of a neural network model during training and validation across multiple epochs. The upper graph illustrates the progression of training and validation accuracy over epochs. As training advances, both accuracies demonstrate an upward trend, indicating the model's improvement in learning and ability to generalize to unseen data. Conversely, the lower graph presents the training and validation loss over epochs. Here, the declining trend in loss values suggests the model's enhanced predictive capabilities, as lower losses correspond to improved performance. The convergence of training and validation metrics signifies the model's effective learning of data patterns and its ability to generalize predictions accurately.

8.1 Model Performance:

The model achieved accuracy of 98.75% on the test dataset with a loss of 0.0510. The accuracy and precision of the model on the training and validation data is 98.69% with 0.03484 loss and 98.46% and 0.0475.

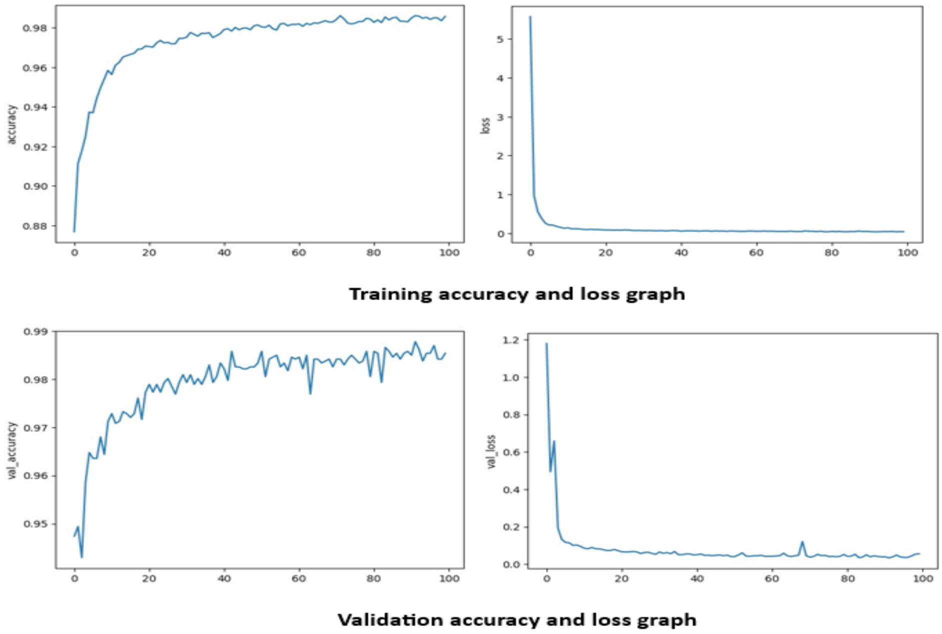

Figure 8.1: Training and Validation Accuracy and Loss Graph

```
78/78 [==============================] - 0s 2ms/step - loss: 0.0510 - accuracy: 0.9875
Test loss: 0.05104377120733261
Test accuracy: 0.9874544739723206
232/232 [==============================] - 0s 2ms/step - loss: 0.0348 - accuracy: 0.9869
train loss: 0.034844476729631424
train accuracy: 0.9869095683097839
78/78 [==============================] - 0s 2ms/step - loss: 0.0475 - accuracy: 0.9846
val date: 0.047500018030405045
val accuracy: 0.9846153855323792
```

Figure 8.2: Accuracy and loss results

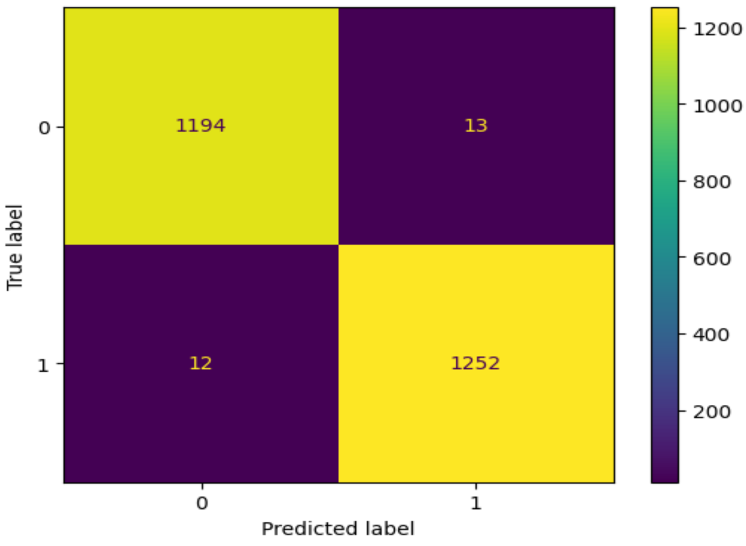

Figure 8.3: Confusion matrix on test data

We ran our model on some of the images in the test dataset. The results show that themodel is able to correctly identify the encroachment in the given area.

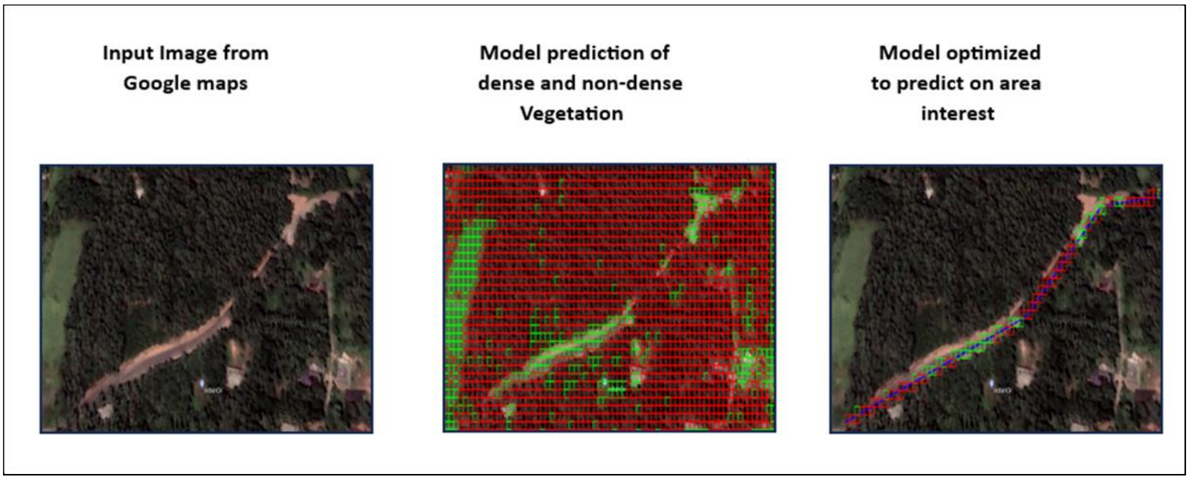

Figure 8.4: Predicted images

The validation of the model was done near Nitte gents hostel the model predictions accurately match with the actual output.

Area near Nitte Gents hostel
Where encroachment was detected
In area of interest

Ground image of the area
where there is encroachment

Figure 8.5: Figure showing encroachment

Area near Nitte Gents hostel
Ground where there is no encroachment
in area of interest

Ground image of the area
where there is no encroachment

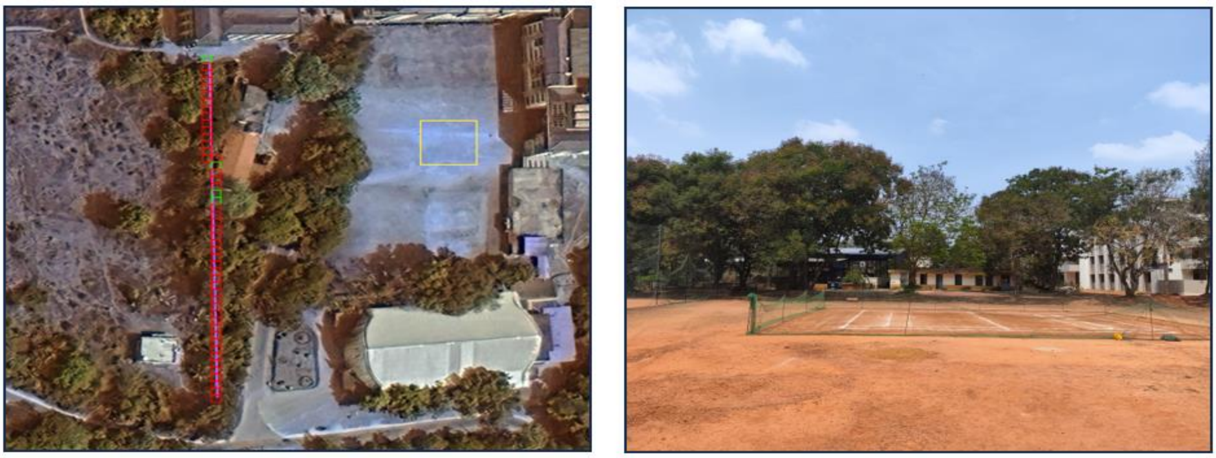

Figure 8.6: Figure without encroachment

8.2 GUI and Webapp Integration:

We have also built a web – based application to take input place latitude and longitude values and it displays the output generated by the model.

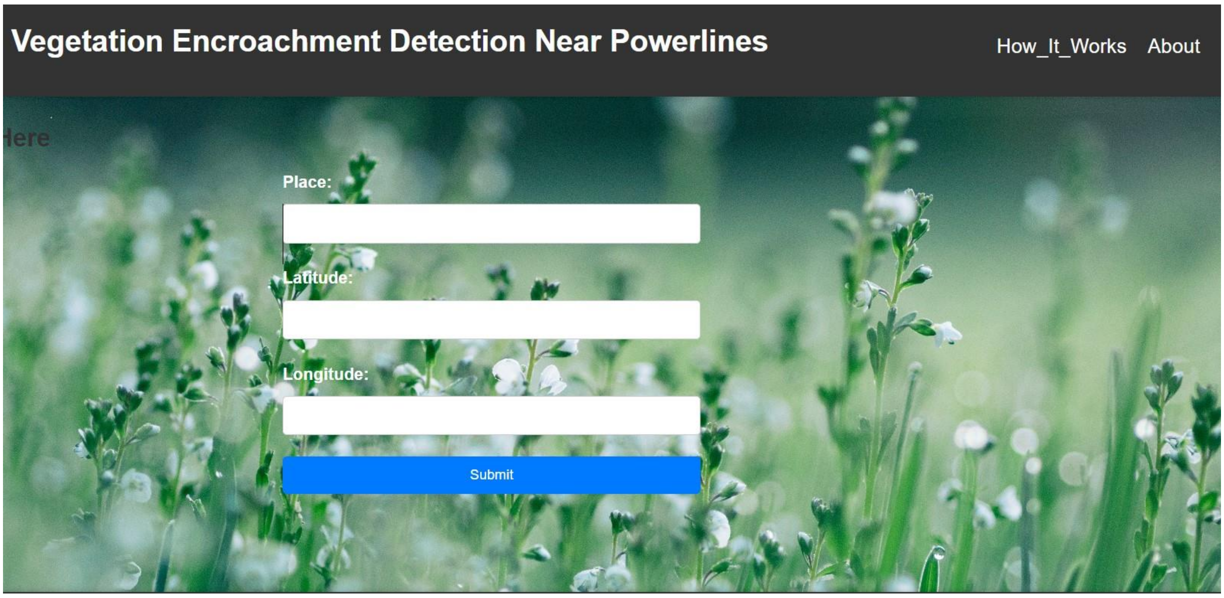

Fig 8.7:Latitude and longitude adding form

Fig 8.10: Prediction webpage

CHAPTER 9
CONCLUSION AND FUTURE WORK

9.1 Conclusion

Our research endeavors have been multifaceted, involving a meticulous examination of diverse strategies, the acquisition of a dataset of paramount importance, and the exploration of future possibilities. Through a rigorous process, we have delved into various approaches, aiming to deepen our understanding of the subject matter at hand. The dataset we meticulously gathered stands as a cornerstone of our research, providing relevant and crucial insights essential for informed decision-making. Furthermore, our exploration of future options has equipped us with invaluable insights, setting the stage for strategic actions in the ever-changing landscape. This comprehensive analysis not only enriches our understanding but also serves as a guiding force for future decisions and strategies. By amalgamating meticulous research methodologies with forward-thinking exploration, we are well-positioned to navigate the dynamic terrain effectively.

9.2 Future Work

Several avenues exist to enhance the project's functionality and reach firstly to develop a model which can detect encroachment for ground based images and integrating and mapping the powerlines segments in all the areas and also analysis of the rate of encroachments in different areas are some of the works that will contribute in improving the model.

REFERENCES

[1] . Yangyu Chen , Jiayuan Lin, Xiaohan Liao(2022) “Early detection of tree encroachment in high voltage powerline corridor using growth model and UAV-borne LiDAR” DOI: 10.1016/j.jag.2022.102740

[2] . Fathi mahdi elsiddig haroun , Siti noratiqah mohamad deros and Norashidah md din (2021) “Detection and Monitoring of Power Line Corridor from Satellite Imagery Using RetinaNet and K-Mean Clustering” DOI 10.1109/ACCESS.2021.3106550

[3] . Levente J Klein ,Wang Zhou ,Hendrik F ,Hamann,Conrad M Albrecht,Carlo ,Siebenschuh,Siyuan Lu,Sharathchandra Pankanti (2019) “N-dimensional geospatial data and analytics for critical infrastructure risk assessment” DOI 978-1-7281-0858-2/19

[4]. Abdul Qayyum, Aamir Saeed Malik, Mohamad Naufal ,Mohamad Saad, Mahboob Iqbal, Rana Fayyaz Ahmad, Tuan Ab Rashid Bin Tuan Abdullah,A hmad Quisti Ramli (2014) “Monitoring of Vegetation Near Power Lines Based on Dynamic Programming using Satellite Stereo Images”

[5] . Rabab Abdelfattah, Xiaofeng Wang, Song Wang (2020) “An Aerial-Image Dataset for Detection and Segmentation of Transmission Towers and Power Lines”

[6] . Zhengrong Li Yuee Liu Ross Hayward Jinglan Zhang Jinhai Cai (2008) “Knowledge-based Power Line Detection for UAV Surveillance and Inspection Systems” DOI 978-1-4244-2582-2/08

[7] . Shuaiang Rong,Liang Du,Zuyi li,Shiwen yu (2020) “Intelligent Detection of Vegetation Encroachment of Power Lines with Advanced Stereovision” DOI 10.1109/TPWRD.2020.3043433

[8] . Nina m. singer, Vijayan k. asari (2021) “A large Scale Benchmark Dataset for Instance Segmentation in Aerial Lidar”

[9] . Seulbi Lee and Youngjib Ham (2023) “Measuring the Distance between Trees and Power Lines under Wind Loads to Assess the Heightened Potential Risk of Wildfire” DOI 10.3390/rs15061485

[10] . Yunping Chen,n , Yang Li , Huixiong Zhang , Ling Tong, Yongxing Cao, Zhihang Xue (2016) “Automatic power line extraction from high resolution remote sensing imagery based on an improved Radon transform” DOI 10.1016/j.patcog.2015.07.004