Runway Detection and Localization in Aerial Images using Deep Learning

A PROJECT REPORT

Submitted by

Muhammed Rishan (MES19CS072)

to

the APJ Abdul Kalam Technological University

in partial fulfillment of the requirements for the award of the Degree

of

Bachelor of Technology

in

Computer Science and Engineering

Department of Computer Science and Engineering
[NAAC Accredited with 'A' Grade]
[B.Tech. Programme Accredited by NBA]
MES College of Engineering Kuttippuram
Thrikkanapuram South P.O, Malappuram Dt, Kerala, India 679582

2022 - 2023

DECLARATION

We, hereby declare that the project report "Runway Detection and Localization in Aerial Images Using Deep Learning", submitted for partial fulfillment of the requirements for the award of degree of Bachelor of Technology of the APJ Abdul Kalam Technological University, Kerala is a bonafide work done under the supervision of Ms.Neena Susan Alex, Assistant Professor, Computer Science and Engineering. This submission represents our ideas in our own words and where ideas or words of others have been included, we have adequately and accurately cited and referenced the original sources. We also declare that we have adhered to ethics of academic honesty and integrity and have not misrepresented or fabricated any data or idea or fact or source in our submission. We understand that any violation of the above will be a cause for disciplinary action by the institute and/or the University and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been obtained. This report has not been previously formed the basis for the award of any degree, diploma or similar title of any other University.

Place: Kuttippuram
Date: 12/06/2023

Muhammed Rishan

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

MES COLLEGE OF ENGINEERING, KUTTIPPURAM

CERTIFICATE

This is to certify that the report entitled **“Runway Detection and Localization in Aerial Images Using Deep Learning”** submitted by **Muhammed Rishan**, to the APJ Abdul Kalam Technological University in partial fulfillment of the requirements for the award of the Degree of Bachelor of Technology in Computer Science and Engineering is a bonafide record of the project work carried out under guidance and supervision. This report in any form has not been submitted to any other University or Institute for any purpose.

Internal Supervisor:

Ms. Neena Susan Alex
Assistant Professor
Dept. of Computer Science and Engg.
MES College of Engineering

Head of the Dept:

Dr. Anil K Jacob
Professor & Head
Dept.of Computer Science and Engg.
MES College of Engineering

Project Coordinator:

Ms. Swathy Sekhar
Assistant Professor
Dept. of Computer Science and Engg.
MES College of Engineering

Project Coordinator:

Ms. Shibina Sharaf.S
Assistant Professor
Dept.of Computer Science and Engg.
MES College of Engineering

ACKNOWLEDGEMENT

First of all I wish to thank Almighty for blessing that made this work a success.I am grateful to ***Dr. Rahumathunza I.***, *Principal, MES College of Engineering, Kuttippuram*, for providing the right ambiance to complete this project. I would also like to extend my sincere gratitude to ***Dr. Anil K Jacob***, *Head of Department, Computer Science and Engineering, MES College of Engineering, Kuttippuram.*

I am deeply indebted to the project coordinators, ***Ms. Swathy Sekhar***, *Assistant Professor, Department of Computer Science and Engineering* and ***Ms. Shibina Sharaf.S***, *Assistant Professor, Department of Computer Science and Engineering* for their continued support.

It is with great pleasure that I express my deep sense of gratitude to my project guide, ***Ms. Neena Susan Alex***, *Assistant Professor, Department of Computer Science and Engineering*, for her guidance, supervision, encouragement and valuable advice in each and every phase.

I would also like to express my sincere thanks and gratitude to all staff members of the department, my friends and family members for their cooperation, positive criticism, consistent support and consideration during the preparation of this work.

Muhammed Rishan

ABSTRACT

The hardest part of any aerial platform's flight is the landing phase. Numerous landing mishaps have occurred as a result of ineffective systems, damaging onboard equipment. Vision-based systems offer a low-cost method of locating landing places by giving detailed textual data. In order to achieve this, this research focuses on precisely identifying and localising runways in aerial images with unkempt terrains. This will subsequently assist aerial platforms, particularly Unmanned Aerial Vehicles (commonly referred to as Drones), in identifying landing targets (i.e., runways) to facilitate automatic landing. The majority of earlier work on runway detection is based on straightforward image processing techniques with numerous assumptions and limitations regarding the precise position of the runway in a given image. In the first phase of this research, runway identification algorithms based on cutting-edge deep learning architectures are developed, and in the second phase, runway localization algorithms using both deep learning and non-deep learning based techniques are developed. The suggested runway detection method consists of two stages, the first of which involves classifying an aerial image to determine whether a runway is present in that particular image. The discovered runways are then localised in the second stage using both traditional line detection methods and more current deep learning models. The runway classification was completed with a score of 0.8 for Intersection-over-Union (IoU) correctness, or about 97 percent.

CONTENTS

Contents Page No.

LIST OF FIGURES

ABBREVIATIONS

UAV	Unmanned Aerial Vehicle
CNN	Convolutional Neural Network
MRCNN	Mask R-Convolutional Neural NetworK
PCNN	Partially Connected Convolutional Neural Network
MPCNN	Multi-Channel Partially Connected Convolutional Neural Network
RBF	Radical Basic Functions
EVS	Environmental Studies
HT	Hough Transform
FPGA	Field-Programmable Gate Array
SRG	Seeded Region Growth
SVS	Structural Variable Selection

%

CHAPTER 1
INTRODUCTION

Due to its employment in jobs that are too risky for manned aerial vehicles to do, unmanned aerial vehicles (UAVs) have grown in popularity over the past few years. Using a range of onboard sensors, including optical pictures, laser scanners, or even synthetic aperture radars, they have been employed for a variety of non-military objectives, including urban planning, inspection, monitoring, surveying, search and rescue, precision agriculture, and many more. A secure landing operation is crucial for the operation of UAVs without hardware damage. To do this, vision-based techniques for UAV landing have been a hot issue in study, in part because they are significantly more suitable for autonomous landing problems than other sensors because they can offer rich textual information at a relatively much lower relative cost. These vision-based technologies can be used in conjunction with traditional control techniques for a stable landing approach. A technique for locating a runway in a picture is known as runway detection. Finding the precise location of the runway in an image is known as localization. In order for UAVs to land and perform self-localization and navigation, runway identification and localization is a crucial task. The vision-based strategy for fixed-wing UAV landing includes a controller to lead the UAV appropriately, a vision-based step for runway recognition, and an alignment of the UAV to the runway. Only the first element of the landing preparation process ramp identification utilising a single onboard camera is the subject of this study.

CHAPTER 2
LITERATURE SURVEY

This section describes many methods that have been used in the past to detect runways for UAV landings as well as for other things like urban planning. The majority of these methods rely on machine learning algorithms, the Hough transform, template matching, and active contours.

2.1 Real time runway detection in satellite images using multi-channel [2]

This study recommends utilising a pulse coupled neural network with Multi-Channel Partially Connected Convolutional Neutral Network [MPCNN] connecting and feeding fields to analyse multispectral pictures. To determine the rapid linkages among neurons with respect to their spectral feature vectors and spatial closeness, pulse based Radial Basis Functions [RBF] units are included into the model neurons of Partially Connected Neutral Network [PCNN], in contrast to the standard Partially Connected Neutral Network [PCNN]. This Multi-Channel Partially Connected Convolutional Neutral Network [MPCNN] can be parallelized on an Field-Programmable Gate Array [FPGA] device to do real-time picture segmentation and edge detection. A modified Hough Transform and landmark feature extraction approach is created to perform airport runway detection in satellite pictures based on the output of the neural circuits. According to experimental findings, the suggested parallel Multi-Channel Partially Connected Convolutional Neutral Network [MPCNN] circuits for segmenting Red Green Blue [RGB] satellite pictures significantly outperform the well-known seeded region growth Seeded Region Growth [SRG] approach in terms of processing speed. Furthermore, the proposed approach maintains a competitive level of detection accuracy.

2.2 A Real-Time Sensor Guided Runway Detection Method for Forward-Looking Aerial Images [3]

The runway detection approach is becoming more and more crucial in both military and civil applications as avionics technology advances. It is difficult to use the existing runway recognition methods for airborne real-time applications since they are primarily designed for satellite images or downward-looking aerial photographs and have a relatively large algorithm complexity. A real-time sensor guided runway detecting approach is provided in this paper. A search region is first established by creating a runway template using sensor data as well as topography data from Environmental Studies [EVS] and Structured Variable Selection [SVS]. The query image is then subjected to a lines extraction within the search region. Finally, the query image and template are subjected to the matching procedure in order to identify the runway area. The proposed strategy offers a high degree of precision and requires 87.7

2.3 Real time method for airport runway detection in aerial images [4]

In this research, it offers a unique approach to successfully detect the airport runway in real time by combining improved chain codes based edge tracking (ICCBET) and Hough transform (HT). In order to reduce the number of pixels and narrow the angle range that HT processes in later, Competence-Based Education and Training [ICCBET] mostly eliminates the thin and wavy lines. Pyramid at the HT stage also significantly lowers the computing cost to meet real-time performance requirements. In order to prevent overflow caused by generating chain lists with dynamic memory allocation, it finally resolve the memory by developing a chain list array for the picture. Experiments on various images demonstrate that our solution overcomes the blur to locate the lines with good accuracy in comparison to the original chain codes based edge tracking and satisfies the real time need with about 23.5 multiples computation reduction Competence-Based Education and Training (OCCBET).

2.4 Airport Detection Base on Support Vector Machine from A Single Image [5]

One of the main transportation goals is the airport. The ability to detect airports is crucial in both the military and civil sectors. In this work, a brand-new approach to identifying airports from a single photograph is suggested. It utilises support vector machine as a classification function and combines texture and form features. First, a Canny edge detector is utilised, followed by the removal of short lines and curves, the detection of long straight lines using the Hough transform, and last, the discrimination of airport runways using a support vector machine. The experimental results show that the suggested automatic airport-detection method works well.

CHAPTER 3
SYSTEM ANALYSIS

3.1 Proposed System

It has been proven in the field of machine learning that CNNs provide us with a more accurate feature representation than other approaches. First Land has been categorised to determine whether or not there is a runway in the image for runway recognition using CNNs. Once it was established that a runway was visible in the image, it was localised using both CNN-based and non-CNN-based techniques. Runway localization and runway detection are the two components of this study.

3.2 Objectives of Proposed System

To build a vision based system using deep learning and image processing techniques to detect runway and localize the runway. Vision based systems provides low cost solution to detect landing sites by providing rich textual information. So the system is trying implement a low cost runway localization system which use only use camera as extra hardware to detect runway.

3.3 Feasibility Study

Feasibility is a test of a system according to workability, impact on the organization's ability to meet user needs and effective use of resources. Use of deep learning in runway detection allows to detect runways without explicitly extract crafted features. The proposed runway detection model has been validated on two datasets. A custom based runway detection dataset. Public remote sensing dataset for aerial image classification which shows that this model can detect any shape of runway with the appropriate training data. Following are the feasibility study employed,

- Technical Feasibility
- Economic Feasibility
- Operational Feasibility

3.3.1 Technical Feasibility

A study of function, performance, and constraints may improve the ability to create an acceptable system. Technical Feasibility is frequently the most difficult area to achieve at the stage of the product engineering process. Considering that are normally associated with the technical feasibility include development risk, resource availability, and technology.

The Technical Feasibility study deals with the hardware as well as software requirements. This project uses a python framework python IDLE, version 3 it is technically feasible.

3.3.2 Economic Feasibility

Desktop applications are widely used nowadays. To use the application the only cost is desktop access. The languages and frameworks used or this project is also free. So this project is economically feasible.

3.3.3 Operational Feasibility

Desktop applications are widely used nowadays. Since the interface is user-friendly, users like admins can easily use. So this project is operationally feasible.

3.4 System Specification

A system requirements specification is a structured collection of information that represents the requirements of a system. Non functional requirements do impose constraints on the design or implementation(such as performance quality standards engineering requirements or design constraints). The software requirements specification lists all necessary requirements if met will offer better optimum software usability.

The system requirements of the project include

- Software specification
- Hardware specification

3.4.1 Software specification

For the proposed system to work properly it requires:

- Operating system : Windows 10
- Tool: Python IDLE
- Python : version 3
- Frontend: Python Tkinter

3.4.2 Hardware specification

- PROCESSOR : i5 or i7
- RAM : Min 8GB
- HARD DISK : 500GB or above

CHAPTER 4
DESIGN AND DEVELOPMENT

4.1 Design Architecture

There are two steps in this aspect of design architecture.

- Training Section
- Prediction Section

4.2 Training Section

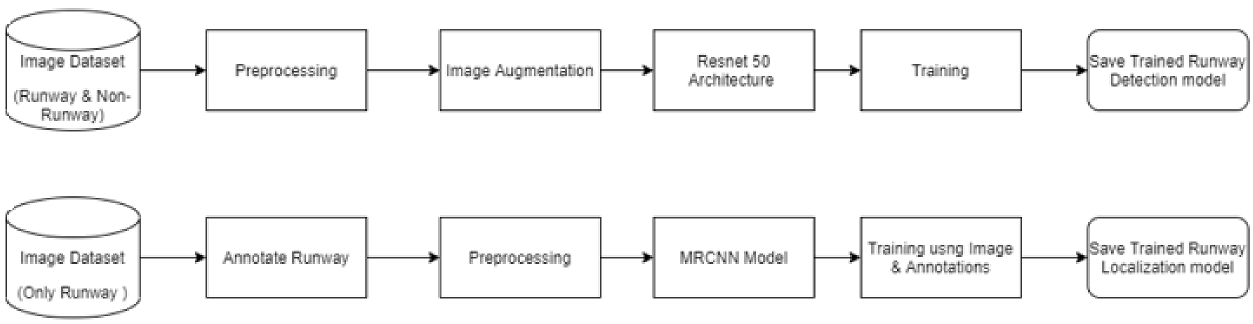

Figure 4.1: Training Stage

4.2.1 Image Dataset

It would have been less appropriate to regard all geographical regions, including roads, woods, mountains, and deserts, as one class for the purposes of binary classification. For this reason, a remote sensing dataset with several classes has been utilised and NWPU-RESISC45 dataset is used which contains 45 classes of aerial images. This dataset, which comprises of satellite photos taken from Google Earth and compiled by specialists, is the largest one currently accessible that is varied enough to apply CNN models. There are 45 classes, and there are 700 photos in each class.

4.2.2 Image Prepossessing

Image preprocessing is the steps taken to format images before they are used by model training and inference. This includes, but is not limited to, resizing, orienting, and color corrections. Image preprocessing may also decrease model training time and increase model inference speed. Preprocessing data is a common step in the deep learning workflow to prepare raw data in a format that the network can accept. Input images have been resized to 224x224 and the only preprocessing performed is mean normalization. Keras models with backend as TensorFlow have been used for feature extraction.

4.2.3 Image Augmentation

Image augmentation creates new data that can be used for model training. In other words, it is the process of enhancing the dataset that is made available for deep learning model training. The practise of creating new, modified copies of the images in the provided image dataset in order to broaden its diversity. Images are nothing more than a 2-dimensional array of numbers to a computer. These figures stand for pixel values, which you can change in a variety of ways to create brand-new, enhanced photos.

4.2.4 Resnet50 Architecture

ResNet-50 is a convolutional neural network that is 50 layers deep. You can load a pretrained version of the network trained on more than a million images from the ImageNet database. The 50layer ResNet architecture includes the following elements, a 7×7 kernel convolution alongside 64 other kernels with a 2 sized stride. A max pooling layer with a 2 sized stride. 9 more layers 3×3,64 kernel convolution, another with 1×1,64 kernels, and a third with 1×1,256 kernels. Here in proposed system, classification layer has been removed to extract 2048 dimensional feature set from images.

4.2.5 Training

After extracting features, a softmax classifier has been trained on these features. This classifier has been implemented using TensorFlow. It works as follows: Weight matrix is initialized using random values based on normal distribution and biases are initialized to zero. Inputs (extracted features) are multiplied with weight matrices and biases are added. Training labels are first converted into one hot encoding sequence (representation of labels as binary vectors). Then loss is calculated by computing cross-entropy and taking average of it across all training examples. The minimum loss is found using gradient descent optimizer. Three classifiers for each model have been trained with different training and testing data and their mean accuracy has been reported in results section for different training ratios.

4.2.6 MRCNN Model

Mask R-CNN is a state of the art model for instance segmentation, developed on top of Faster R-CNN. Faster R-CNN is a region-based convolutional neural networks , that returns bounding boxes for each object and its class label with a confidence score. The goal is to localise the runway so that its borders may be extracted along with it. Bounding boxes provide us with a portion of the image that contains the necessary object. Even now, we are unable to retrieve the actual object. A segmentation algorithm is required to extract the object along with its borders. Each pixel in segmentation is given a class. It is also known as pixel level categorization because it determines whether or not each pixel falls within a specific class. The needed object is given a pixel-by-pixel mask, and the model learns by calculating the difference between the ground truth mask and the expected mask. To start, bounding boxes can be used to enclose the area of the image that will be subject to pixel-by-pixel categorization. Resnet50 serves as its backbone architecture.

4.3 Prediction Stage

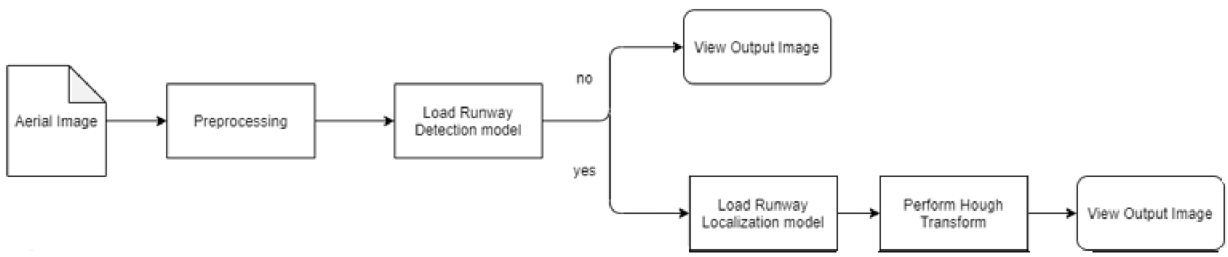

Figure 4.2: Prediction Stage

4.3.1 Aerial Images

An aerial photograph, in broad terms, is any photograph taken from the air. Normally, air photos are taken vertically from an aircraft using a highly-accurate camera. Aerial imagery is an indispensable tool for topographical mapping and the interpretation of places, objects, and features. The remote sensing method collects vital information that can be used for land use, agricultural management, forestry, conservation, urban planning and more.

4.3.2 Hough Transform

The Hough transform is a popular feature extraction technique that converts an image from Cartesian to polar coordinates. Any point within the image space is represented by a sinusoidal curve in the Hough space. In this method, runway photos from the chosen dataset were first turned into grayscale images. Then edges were extracted using Canny edge detection. The Canny algorithm has been used to find edges in the image with a 1:3 hysteresis threshold ratio.

4.4 Use Case Diagram

A use case diagram is used to represent the dynamic behavior of a system. It encapsulates the system's functionality by incorporating use cases, actors, and their relationships. It models the tasks, services, and functions required by a system/subsystem of an application. It depicts the high-level functionality of a system and also tells how the user handles a system.

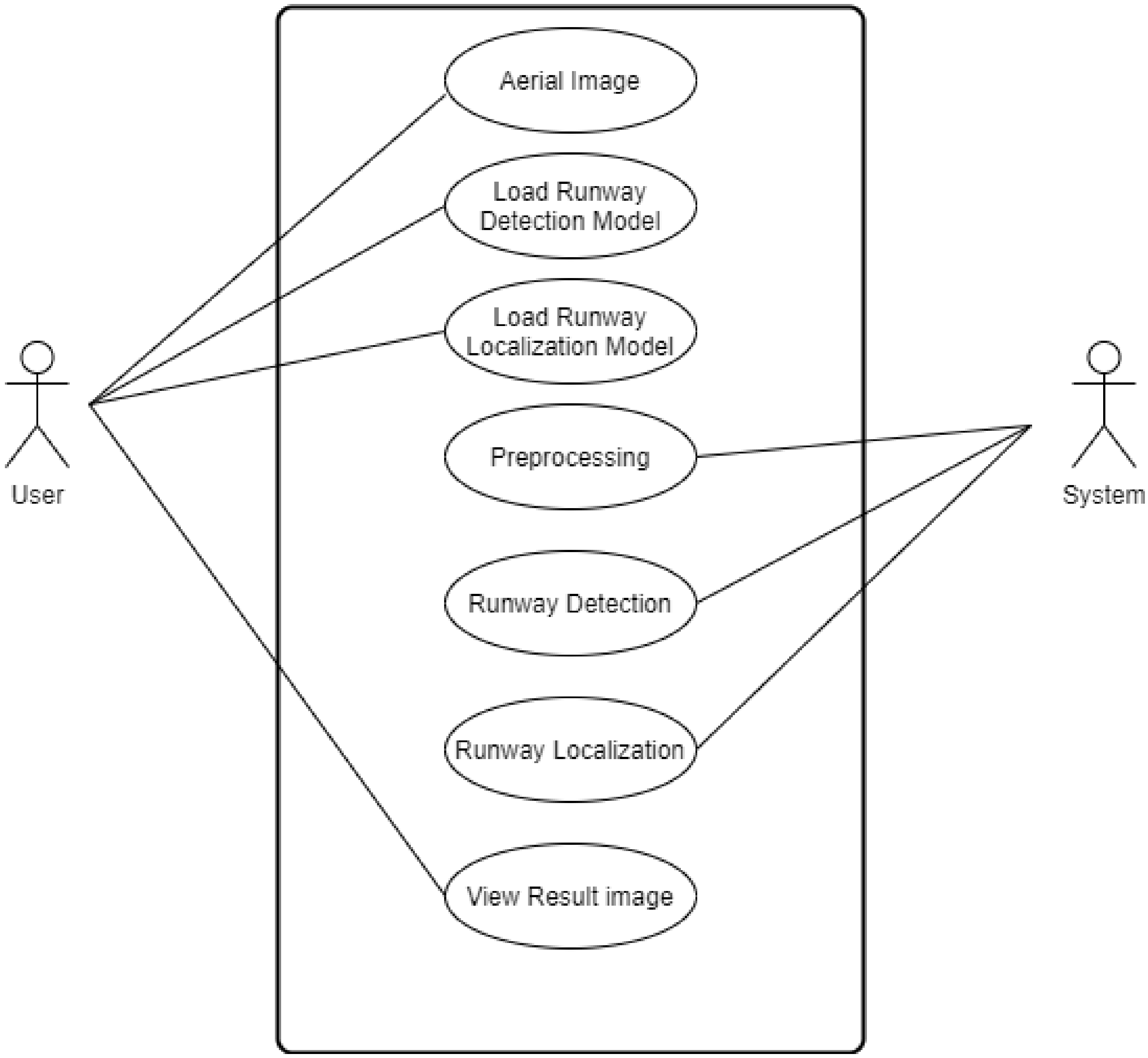

Figure 4.3: Use Case Diagram

CHAPTER 5
METHODOLOGY

5.1 Runway Detection

The following phases make up the runway detection model implementation that has been finished.

- Image Pre-processesing
- Data Augmentation Initialization
- Train-Test Splitting
- Runway Detection Model Creation

Remote sense image scene classification is carried out using the NWPU-RESISC45 dataset. RESISC45 dataset is a publicly available benchmark for Remote Sensing Image Scene Classification (RESISC), created by Northwestern Polytechnical University (NWPU). This dataset contains 31,500 images, covering 45 scene classes with 700 images in each class.

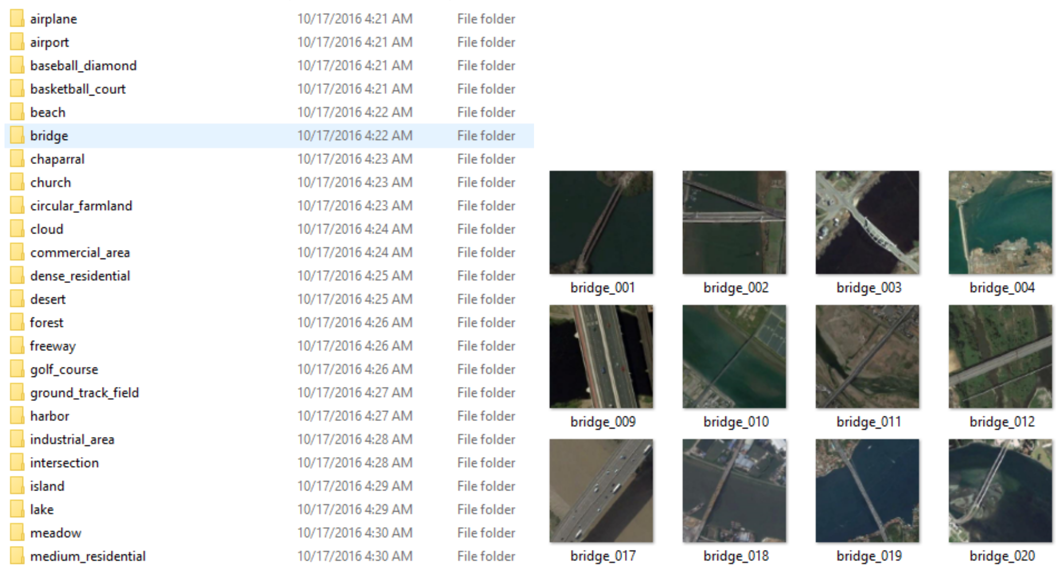

Figure 5.1: Some NWPU-RESISC45 Dataset folders and Images[Bridges] taken from the dataset folder

5.1.1 Image Pre-processing

The dataset [NWPU- RESISC45] should be loaded to obtain each image. Each folder [label] from the dataset will be iterated in order to produce the appropriate images from the labels. Read and resize the iterated images, and update the [data, label] lists. Labels should be converted to integers from their existing string form.

5.1.2 Train-Test Splitting

Divide the data into two groups: the training set and the testing set. The model will be trained on the training set, and it will be tested on the testing set. Data of 25percent will be tested, and 75percent will be used for training.

5.1.3 Data Augmentation Initialization

In data analysis, procedures called ”data augmentation” are used to expand the amount of data by adding slightly changed versions of either existing data or brand-new synthetic data that is derived from existing data. Here, the image quality should be improved by extending the rotation range, zoom range, width shift range, height shift range, shear range horizontal flip, and fill mode. The images will then be optimised.

5.1.4 Runway Detection Model Creation

5.1.4.1 ResNET-50 Model

A 50-layer convolutional neural network is called ResNet-50 (48 convolutional layers, one MaxPool layer, and one average pool layer). Create a model using ResNet-50 without a classifier layer and designate the loaded layers as untrainable. Add more layers to the classifier, such as the flatten layer, the class 1 dense layer with 1024 neurons, the class 2 dense layer with 512 neurons, and the output dense layer with 36 neurons. The model must be summarised in order to determine which layer has occurred and to collect the input from the layers. Create a runway detection model architecture by compiling the model.

5.1.4.2 Train the Model

The dataset for the training purpose will be passed. The split testing dataset will be provided for the validation to be evaluated.Thus, the model resnet50 in the dataset will be trained. The trained model with the best validation accuracy will be saved for future predictions. Here, a graph is created that plots the amount of loss and accuracy in the trained model. Train loss, Train accuracy, Validation loss, Validation accuracy are shown with the Epoch in (x-axis) and loss/accuracy in (y-axis). Validation accuracy on training ratio of 80 percent has been reported to be 97.33 percent and test accuracy on training ratio of 80 percent has been reported to be 96.63percent. To evaluate the model on target class runway, precision and recall has been calculated for class runway. For training ratio of 80 percent, precision and recall has been calculated as 94.44 percent and 97.14 percent respectively. This means that capability of model to correctly classify a runway image as runway is a little more as compared to capability of a model to not classify a non-runway image as runway.

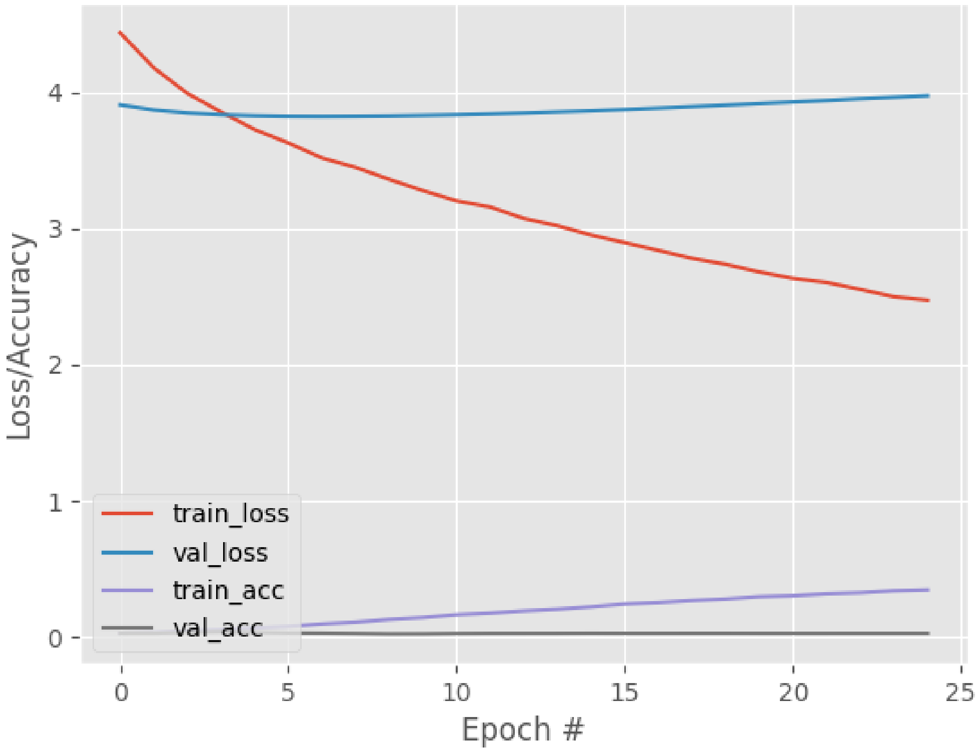

Figure 5.2: Training loss and accuracy on dataset

5.2 Runway Localization

5.2.1 Training Model

5.2.1.1 MaskR-CNN Model

Mask R-CNN is a popular deep learning instance segmentation technique that performs pixel-level segmentation on detected objects. The Mask R-CNN algorithm can accommodate multiple classes and overlapping objects. It creates a pre-trained Mask R-CNN network using the MaskR-CNN object. Here the main objective is that if there is runway only then there is purpose of localization, so there is no need of training from the beginning. A dataset must be prepared for training purposes. The dataset includes a relevant annotation for each images in it. By marking the images that will be placed in the dataset for each image that has a runway, the appropriate coordinates will be trained. The only images that may be used to train MaskR-CNN model and localize an object are those that are kept in the dataset with the correct annotation. Another pre-trained models are publically available i..e, MaskR-CNN COCO h5 weights file contains pre-rained weights for the MaskR-CNN model on the COCO dataset. This weights as a starting point transfer learning provides a solid foundation for the custom runway localization task, helping the model learn faster, generalize better, and achieve good performance even with limited data. So here MaskR-CNN is customized by training where ResNet-50 is used as the backbone. Load the trained runway dataset and save the trained model for the future prediction.

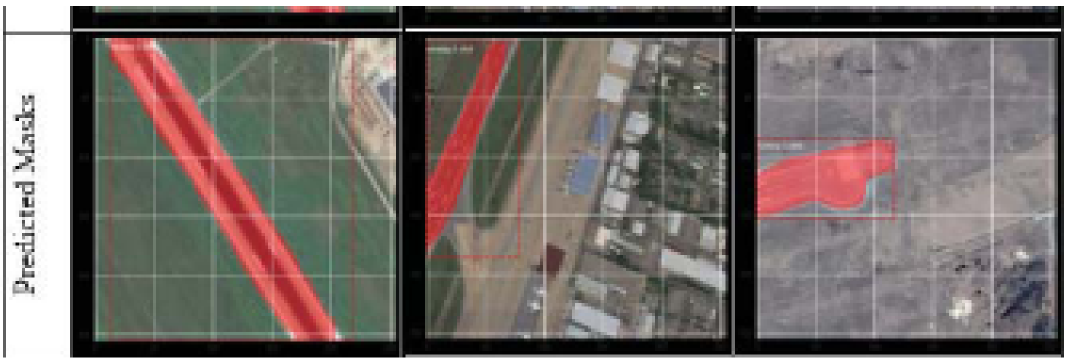

Figure 5.3: Mask R-CNN results on customized dataset.

5.2.2 Prediction Model

Trained runway detection model using ResNet-50 will be loaded. The loaded images will read and the read images will be resized into (256,256) format, then the resized images will be converted into array format. Then the resized array model will be set to the prediction stage. The loaded images will predict if the there is runway or not.

5.2.2.1 Runway not detected:

If runway is not detected then it will stop by without showing the result. Here runway not detected so no prediction of result.

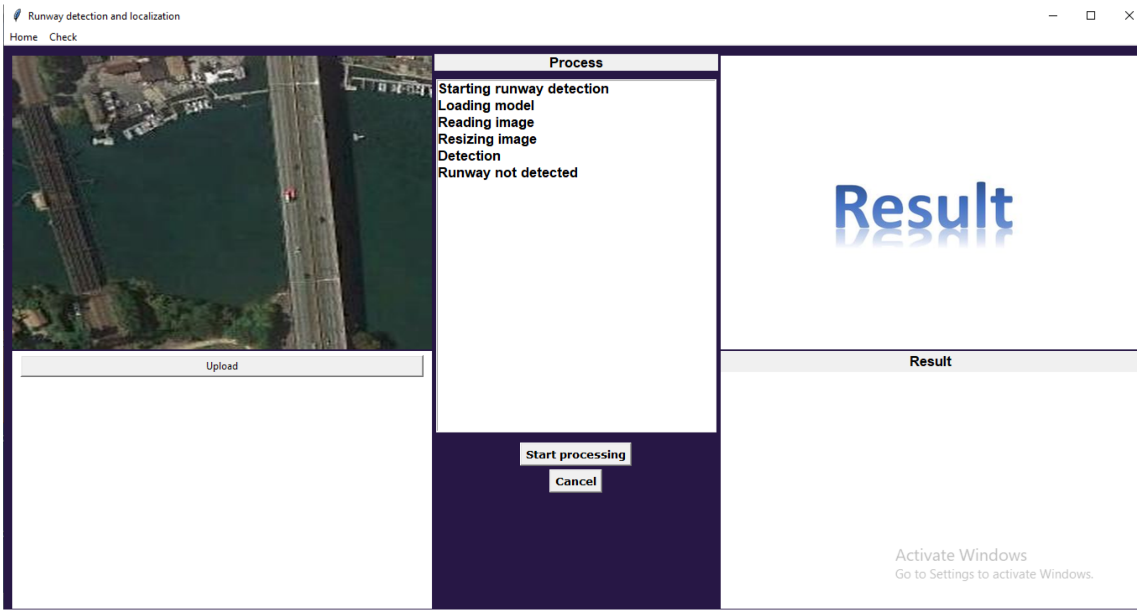

Figure 5.4: Blank result prediction

5.2.2.2 Runway detected :

If runway is detected then the trained Mask-RCNN model image will be loaded. Then segmentation will be performed on the loaded images and the images will be saved.

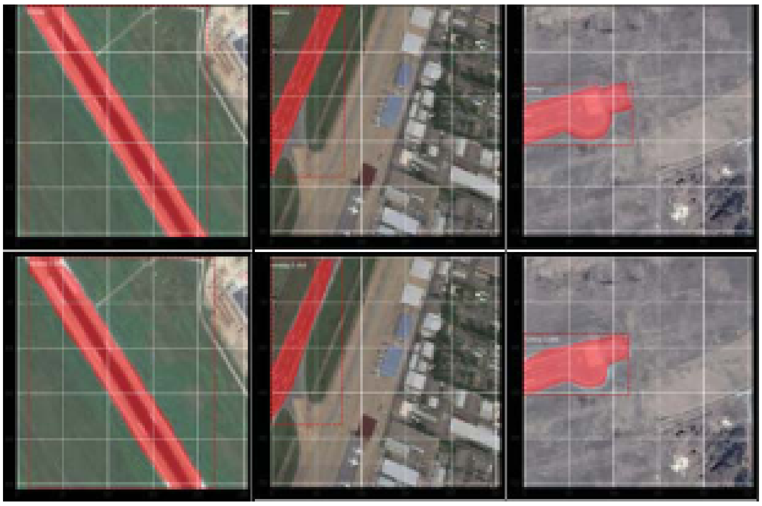

Figure 5.5: Runway detected and performing Localization

5.2.2.3 Hough Transform :

The Hough transform is a feature extraction technique used in image analysis, computer vision, and digital image processing. The purpose of the technique is to find imperfect instances of objects within a certain class of shapes by a voting procedure Here the edges and coordinates of the loaded runway model will be stored then will be saved to perform localization.

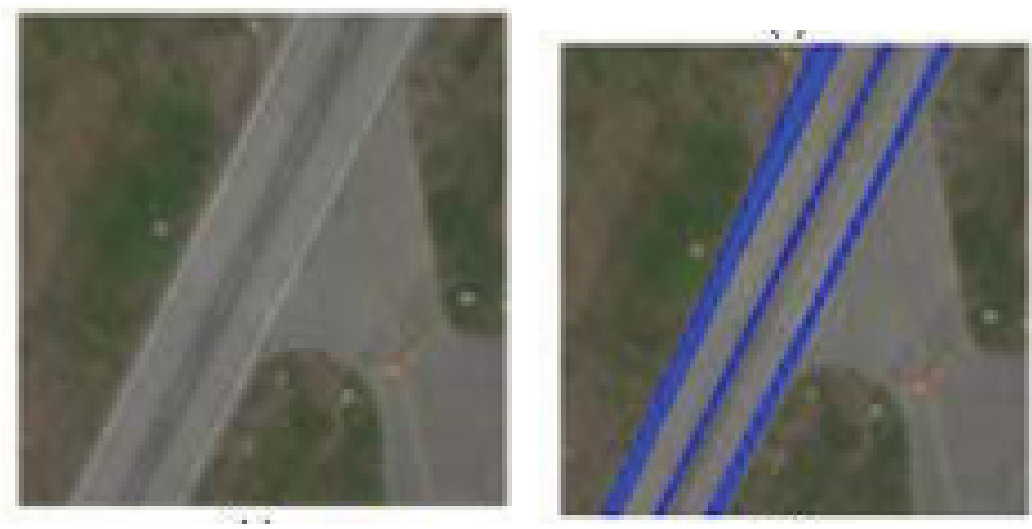

Figure 5.6: Original image, Result of applying HT

After the runway is detected, it will perform segmentation and it will apply hough transform to the loaded Mask-RCNN images. Here the runway is detected and localized by showing the predicted result.

Figure 5.7: Accessed through Anaconda Prompt where the installed environment gets activated and directly goes access to the desktop application.

CHAPTER 6
IMPLEMENTATION

It is basically a desktop application which is accessed through the Anaconda prompt. It is a build-in prototype model, where different sets of libraries are used which is installed in a virtual environment. When this environment gets activated only then the user can run anaconda prompt. It may not support the libraries if the user want to make this application an executable file.

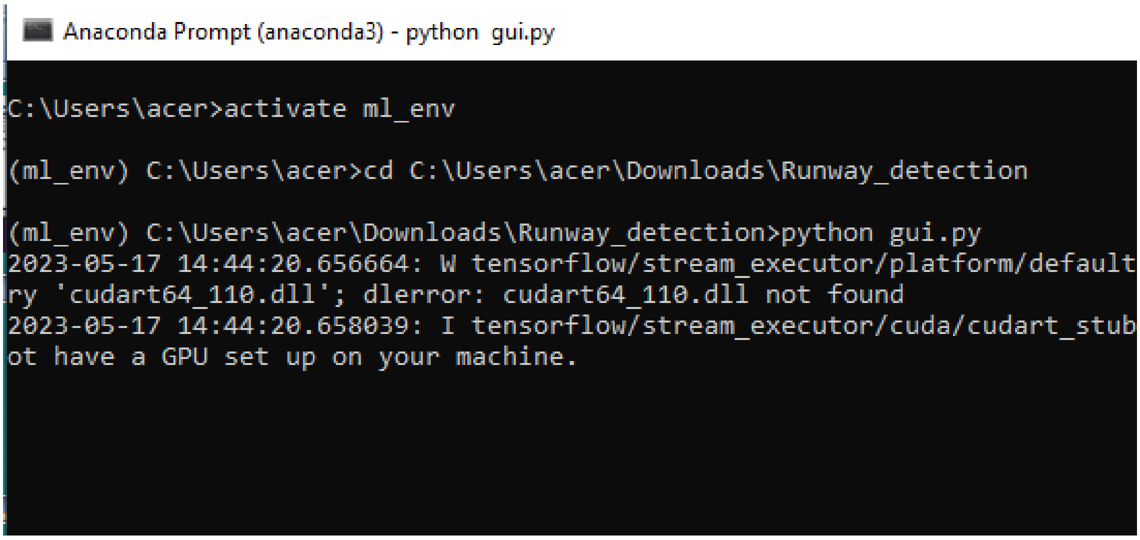

Figure 6.1: Accessed through Anaconda Prompt where the installed environment gets activated and directly goes access to the desktop application.

6.1 Working:

6.1.1 Login page:

Accessed through Anaconda Prompt where the installed environment gets activated and directly goes access to the desktop application

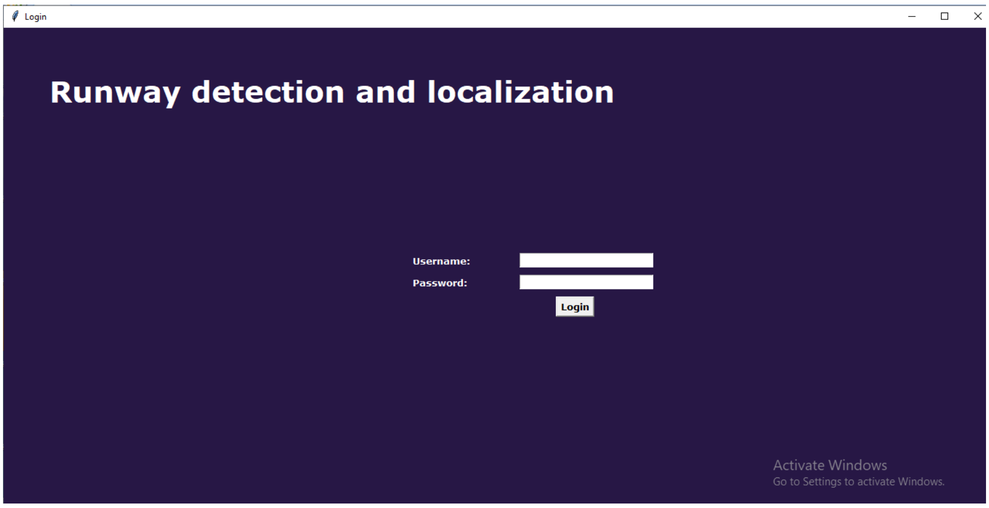

Figure 6.2: Login Page

6.1.2 Home Page:

This is basically the home page. From this page the user can make access to the runway detection and localization

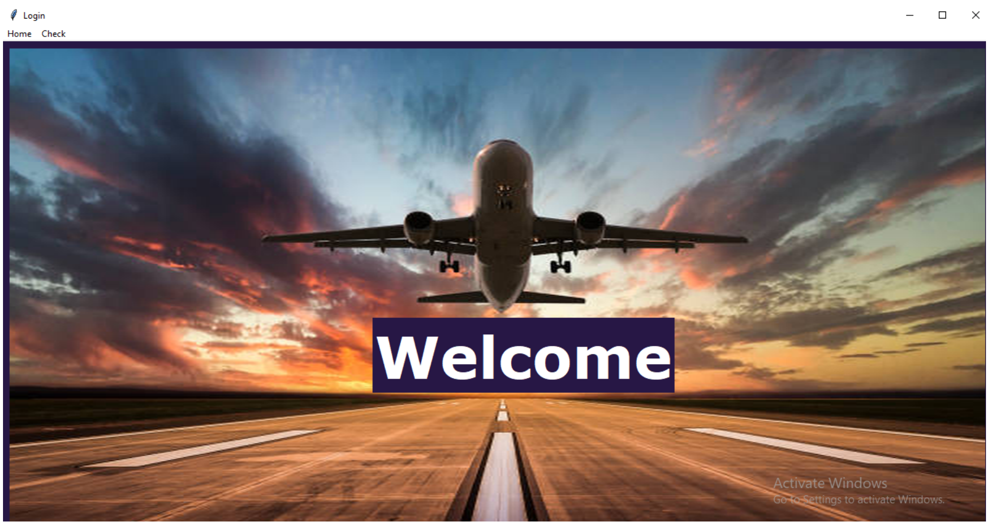

Figure 6.3: Home Page

6.1.3 Check Point Access

The user can access the check menu from the menu bar from the login page. It will make the user access to the runway detection and localization.

Figure 6.4: Check Point Access

6.1.4 Upload Button Access

The runway images from the dataset can be uploaded by the user. The images can be any aerial images from the dataset.

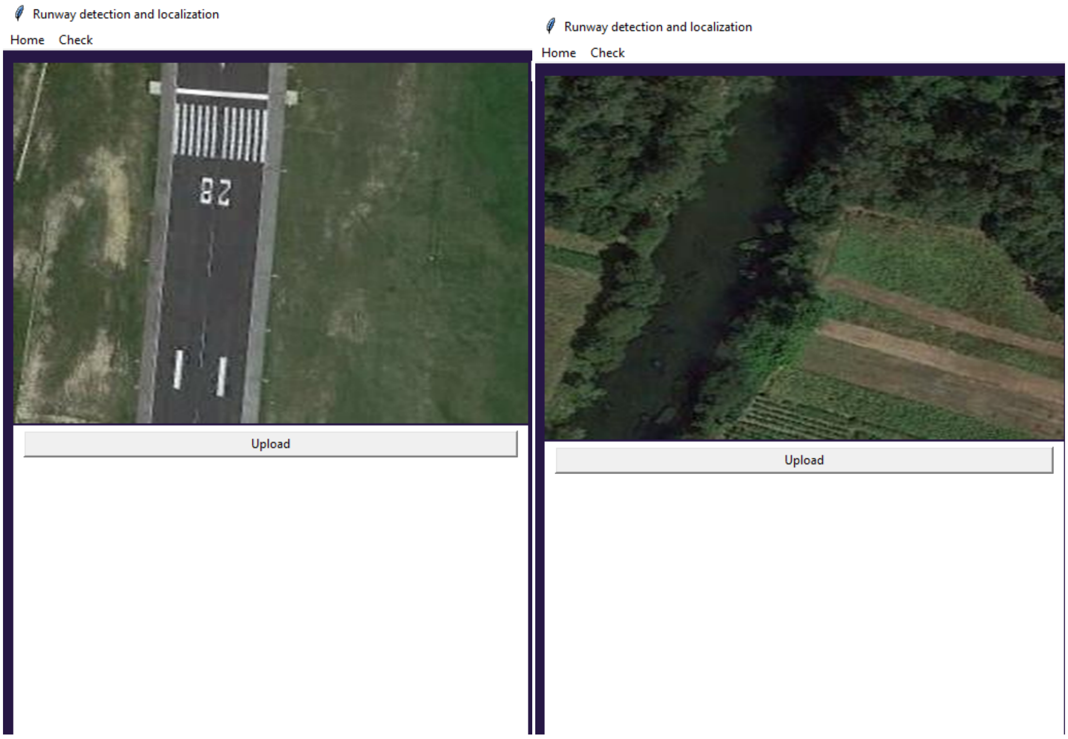

Figure 6.5: Two forms of aerial images from the dataset that where uploaded

6.1.5 Start Processing

After image is uploaded, then processing will be started Processing is of two forms mainly when the predictions made without runway detection and when the predictions made when the runway is detected.

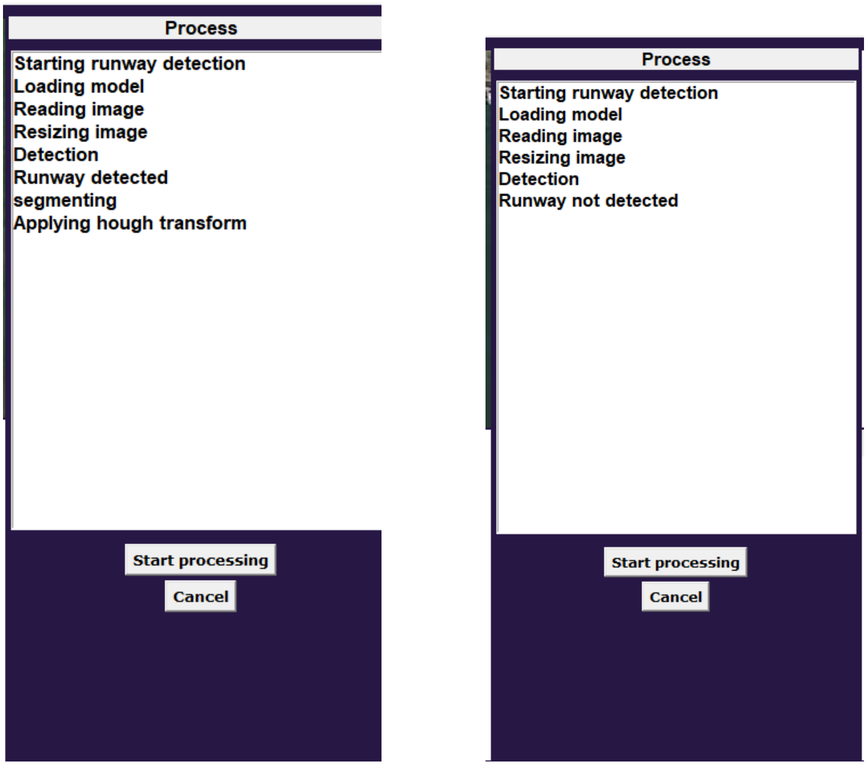

Figure 6.6: runway not detected predictions and runway detected predictions.

6.1.6 Result

After the upload and processing the predicted resulted will be shown in two forms The images where the runway detected will be localized by performing some sort of actions. And the images where the runway is not detected will stop processing, and result side will be blank .

Figure 6.7: The runway detected and the localized image.

CHAPTER 7
CODING

7.1 Python

Python is a general-purpose interpreted, interactive, object-oriented, and high-level programming language. Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently whereas other languages use punctuation, and it has fewer syntactical constructions than other languages. Python is Interpreted Python is processed at run-time by the interpreter. Users do not need to compile the program before executing it. This is similar to PERL and PHP. Python is an Interactive User who can sit at a Python prompt and interact with the interpreter directly to write programs. Python is Object-Oriented Python supports Object-Oriented style or technique of programming that encapsulates code within objects. Python is a Beginner's Language Python is a great language for beginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.

7.2 Anaconda

Anaconda is a distribution of the Python and R programming languages for scientific computing (data science, machine learning applications, large-scale data processing, predictive analytics, etc.), that aims to simplify package management and deployment. The distribution includes data-science packages suitable for Windows, Linux, and macOS. Package versions in Anaconda are managed by the package management system conda. This package manager was spun out as a separate open-source package as it ended up being useful on its own and for things other than Python. There is also a small, bootstrap version of Anaconda called Miniconda, which includes only conda, Python, the packages they depend on, and a small number of other packages.

7.3 Visual Studio Code

For authoring, debugging, and producing code as well as releasing apps, Visual Stu- dio is a helpful tool. Beyond the standard editor and debugger included in most IDEs, Visual Studio comes with a number of extra tools to facilitate the software develop- ment process, including compilers, code completion tools, graphical designers, and a number of other features. The suggested system employs the Visual Studio platform to provide a web form for real-time data monitoring, remote equipment management via the cloud, and report authoring

CHAPTER 8
CONCLUSION

The method for doing runway recognition using aerial photos obtained from an on-board vision sensor is presented in this research. This paper's study is the first phase towards UAV landing, which includes the identification and localization of runways. The goal of this research was to develop an accurate runway detecting model. The majority of earlier studies relied on techniques that weren't based on machine learning and made numerous assumptions about the location of the runway in the image. Deep learning is used to detect runways so that hand-crafted features don't need to be explicitly extracted. The proposed runway detection model has been verified on two datasets, including a private remote sensing dataset for aerial picture classification and a custom runway detection dataset, demonstrating that this model can detect any shape of runway with the right training data. To determine if a runway is present or not, the land was first categorised. Next, a runway detection model was used to extract the runway from a picture. Combining these two methods improves accuracy. This can be used for UAV landing with the appropriate hardware. When the runway has been successfully retrieved, the extracted runway can be utilised to align the UAV with the runway. The proposed runway detection model's effectiveness was validated by its respectable IOU of 0.8.

REFERENCES

[1] J. Akbar, M. Shahzad, M. I. Malik, A. Ul-Hasan and F. Shafait, *"Runway Detection and Localization in Aerial Images using Deep Learning,"* 2019 Digital Image Computing: Techniques and Applications (DICTA), 2019, pp. 1-8.

[2] H. Zhuang, K. S. Low, *"Real time runway detection in satellite images using multi-channel PCNN,"* 2014 9th IEEE Conference on Industrial Electronics and Applications, Hangzhou, 2014, pp. 253-257.

[3] J. Wang, Y. Cheng, J. Xie, W. Niu, *"A Real-Time Sensor Guided Runway Detection Method for Forward-Looking Aerial Images.",* 2015 11th International Conference on Computational Intelligence and Security (CIS), 2015, pp. 150-153.

[4] N. Di, M. Zhu, Y. Wang, *"Real time method for airport runway detection in aerial images,"* 2008 International Conference on Audio, Language and Image Processing, Shanghai, 2008, pp. 563-567.

[5] Y. Qu, C. Li, N. Zheng, 2005 5th International *"Airport Detection Base on Support Vector Machine from A Single Image,"* on Information Communications amp; Signal Processing, Bangkok, 2005, pp. 546-549.

[6] G. Chen B. Cai, Z. Jiang, H. Zhang, D. Zhao, Y. Yao, *""Airport Detection Using End-to-End Convolutional Neural Network"* with Hg, J. Han, X. Lu

[7] G. Cheng, J. Han, X. Lu *"Remote sensing image scene classification: Benchmark and state of the art."* Proceedings of the IEEE, vol. 105, issue 10, 2017.

[8] P. T. G. Jackson, C. J. Nelson, J. Schiefele, B. Obara, *“Runway detection in High Resolution remote sensing data,”* 2015 9th International Symposium on Image and Signal Processing and Analysis (ISPA), Zagreb, 2015, pp. 170-175.

[9]] Ö. Aytekin, U. Zöngür and U. Halici, *”Texture-Based Airport Runway Detection,”* IEEE Geoscience and Remote Sensing Letters, vol. 10, no. 3, pp. 471-475, May 2013.