Runway Detection and Localization in Aerial Images Using Deep Learning

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

Nived A (MES19CS088)
Riza Rahmath Zakir (MES19CS098)
Rinoon Sadiq (MES19CS095)
Muhammed Rishan (MES19CS072)

CONTENT

- INTRODUCTION
- SYSTEM MODEL
- LITERATURE SURVEY
- DESIGN ARCHITECTURE
- IMPLEMENTATION
- WORKING
- FEASIBILITY STUDY
- REQUIREMENTS
- REFERENCE

1. INTRODUCTION

Objective

- 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 we are trying to implement a low cost runway localization system which only use camera as extra hardware to detect runway.

2. SYSTEM MODEL

Creating runway detection model

- All the aerial images (45 classes) from dataset are loaded and given corresponding labels
- All images are resized to specific size
- Image augmentation techniques are applied by choosing random images from dataset
- To increase efficiency of runway detection in bad weather.
- Rotation, translation, random noise insertion, brightness adjustment are applied.
- Prepare resnet50 architecture by changing last dense layer to 45 outputs
- Load pre trained weights for resnet50
- Fine tune resnet50 with aerial images (training)
- Save model

Creating Runway localization model

- Load all images of runway from dataset
- Annotate runway in all images using labels package of python
- Output will be json documents with runway annotation for each image
- Load MRCNN model
- Train MRCNN model using image and corresponding json files
- Save MRCNN model

LITERATURE SURVEY

Real time runway detection in satellite images using multi-channel

AUTHORS : Hualiang Zhuang; Kay Soon Low

- This paper proposes a pulse coupled neural network with multi-channel (MPCNN) linking and feeding fields for multispectral image processing.
- This MPCNN can be implemented in parallel on a FPGA chip to perform real-time image segmentation and edge detection.
- Experimental results show that the proposed parallel MPCNN circuits drastically improve the processing speed over the popular seeded region growing (SRG) algorithm for segmentation of RGB satellite images.

A Real-Time Sensor Guided Runway Detection Method for Forward-Looking Aerial Images

AUTHORS : Jingzhe Wang, Yue Cheng

- The existing runway detection methods are mostly intended for satellite images or downward-looking aerial photos and the algorithm complexity is relatively high, which makes it hard to apply them to airborne real-time applications.
- In this paper, a real-time sensor guided runway detection method is presented. First of all, with the help of sensor data and terrain data from EVS and SVS, a runway template is generated and a search region is therefore determined.
- Then within the search region, a lines extraction on query image is performed. And finally the matching process is applied to the query image and template to detect the runway area.

Real time method for airport runway detection in aerial images

AUTHORS : Nan Di; Ming Zhu; Yinan Wang

- In this paper, they propose a novel method that combines improved chain codes based edge tracking (ICCBET) with Hough transform (HT) to successfully detect the airport runway in real time.
- Furthermore, through using pyramid during HT stage, the computation cost is reduced considerably to satisfy real time performance.
- Finally, they fix the memory by designing a chain list array for the image to avoid overflow arise from building chain list with memory dynamic allocation.

Airport Detection Base on Support Vector Machine from A Single Image

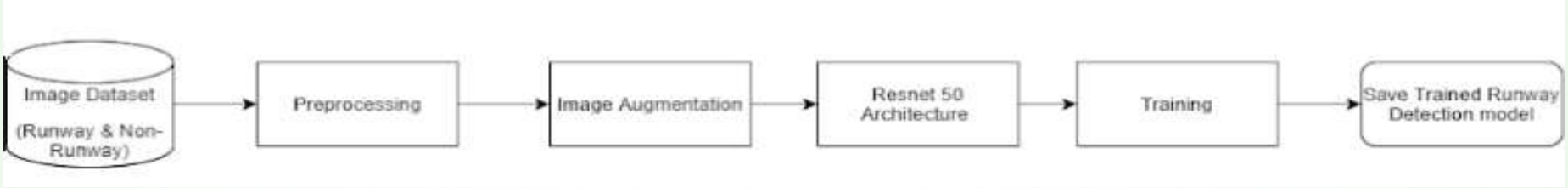
AUTHORS : Yanyun Qu; Cuihua Li; Nanning Zheng

- A novel method to detect airports from a single image is proposed in this paper.
- It combines texture features with shape features, and uses support vector machine as a classification function.
- Canny edge detector is firstly used, then short lines and curves are removed, and long straight lines are detected by Hough transform, at last the airport runways are discriminated by support vector machine.
- The experimental results demonstrate the efficacy of the proposed automatic airport-detection method.

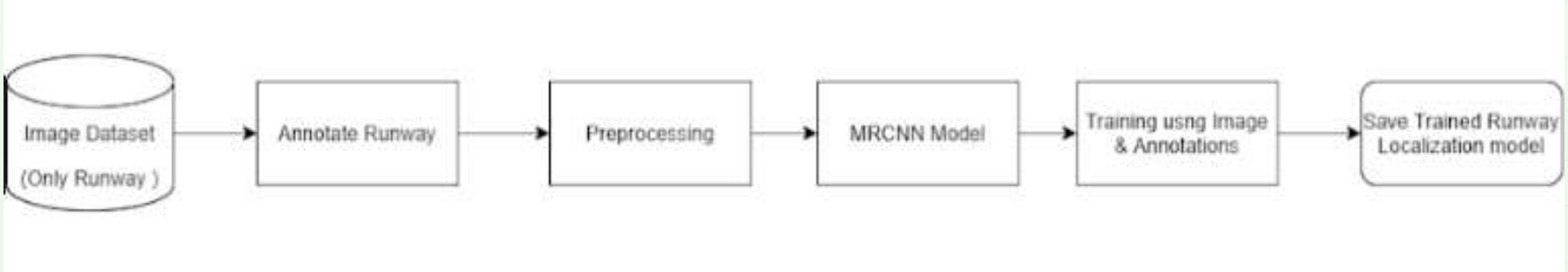
DESIGN ARCHITECTURE

TRAINING STAGE :

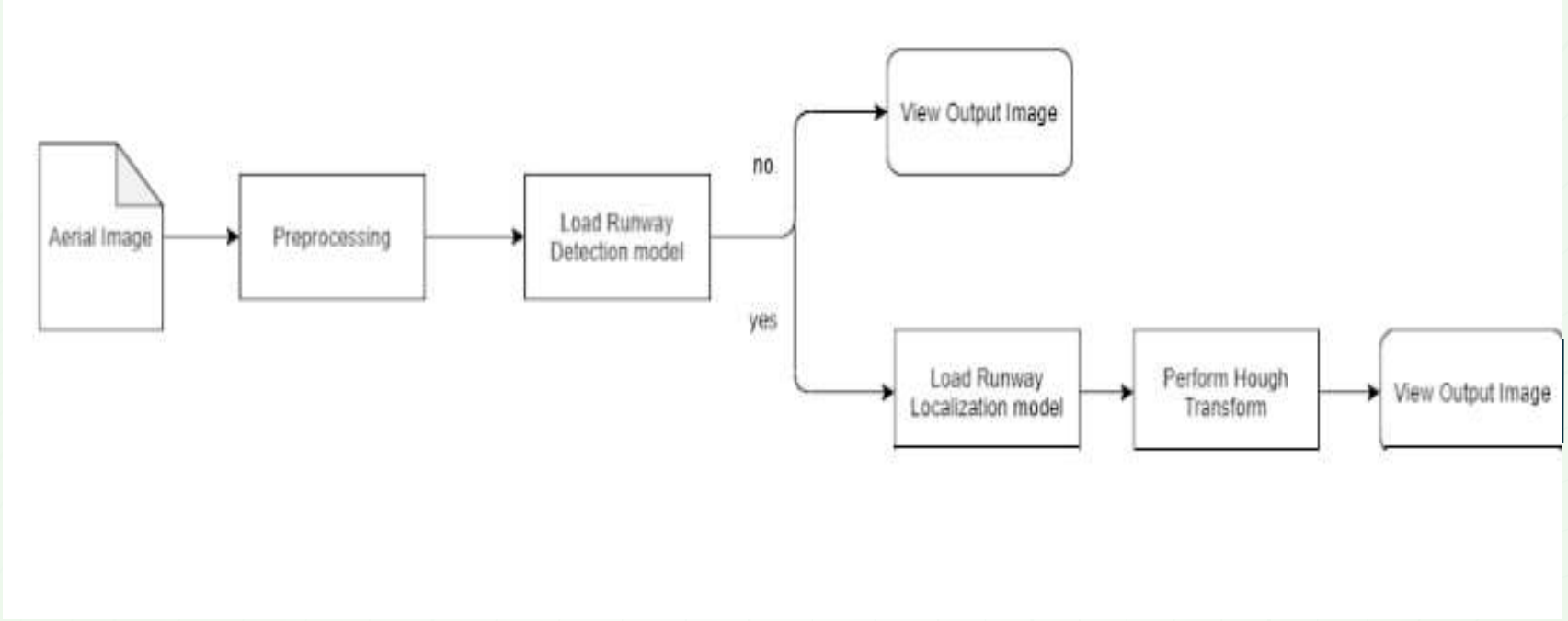
- RUNWAY DETECTION



- RUNWAY LOCALIZATION



PREDICTION STAGE :



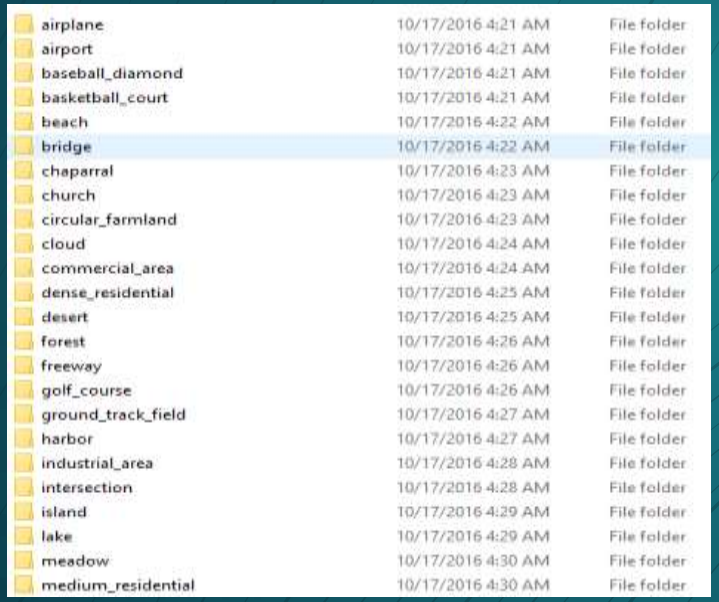
IMPLEMENTATION

RUNWAY DETECTION

Runway detection has the following phases:

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

- Here, remote sense image scene classification is carried out using the NWPU-RESISC45 dataset.
- There are 700 photos in each of the 45 scene classes in this dataset's 31,500 total photographs.



Some NWPU-RESISC45 Dataset folders

Images[Runway] taken from the dataset folder

➢ IMAGE PRE-PROCESSES

- Each image should be loaded from the dataset [NWPU- RESISC45].
- In order to create the relevant images from the labels, each folder [label] from the dataset will be iterated.
- Update the [data, label] lists, read, and resize the iterated images.
- Encode the labels [which are currently strings] as integers.

➢ TRAINTEST SPLITTING

- Partition the data into two splitting state, Training set and Testing set.
- Training set will train the model and the testing set will evaluate the model.
- 75% of data will be trained and 25% of the data will be tested

➢ DATA AUGMENTATION INITIALIZATION

- Data augmentation in data analysis are techniques used to increase the amount of data by adding slightly modified copies of already existing data or newly created synthetic data from existing data.
- Here rotation range, zoom range, width shift range, height shift range, shear range horizontal flip, fill mode should need to augmented for better image.
- Finally the images will be optimized.

➢ RUNWAY DETECTION MODEL CREATION

1. ResNet-50 Model

- ResNet-50 is a 50-layer convolutional neural network (48 convolutional layers, one MaxPool layer, and one average pool layer).
- Using ResNet-50 create a model without a classifier layer and mark the loaded layers as non-trainable.
- Add new classifier layers like flatten layer, class1 dense layer with 1024 neurons, class2 dense layer with 512 neurons, output dense layer with 36 neurons.
- To understand which layer has occurred and to obtain the input from the layers, the model must be summarized.
- Compile the model so that runway detection model architecture is created.

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).
- In the graph, accuracy is in range between 1 and 0 where 1 denotes 100 percent accuracy.
- When the Epoch increases there will be decrease in the loss.

RUNWAY LOCALIZATION

1. Training the model

- Mask-RCNN 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 Mask-RCNN 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 Mask-RCNN 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, Mask-RCNN COCO h5 weights file contains pre-rained weights for the Mask-RCNN 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 Mask-RCNN 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.

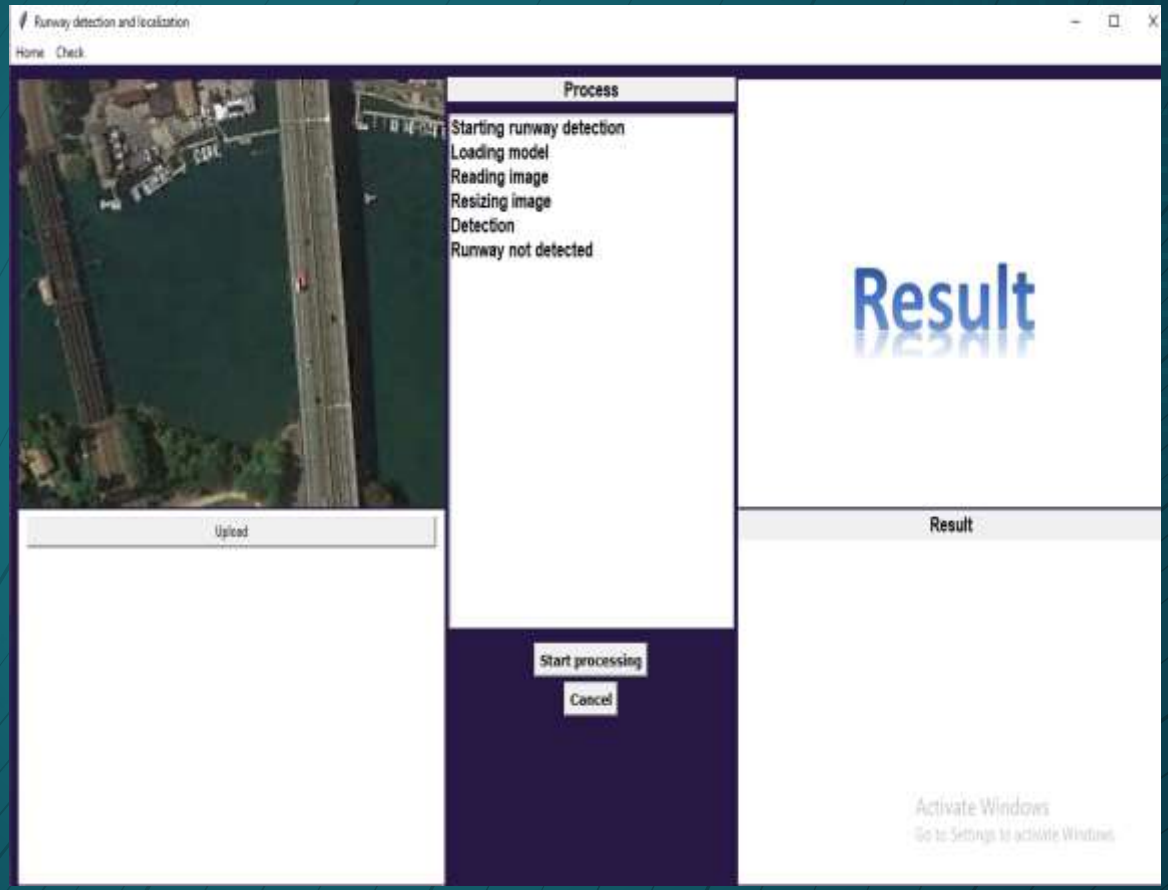
2. Prediction model

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.

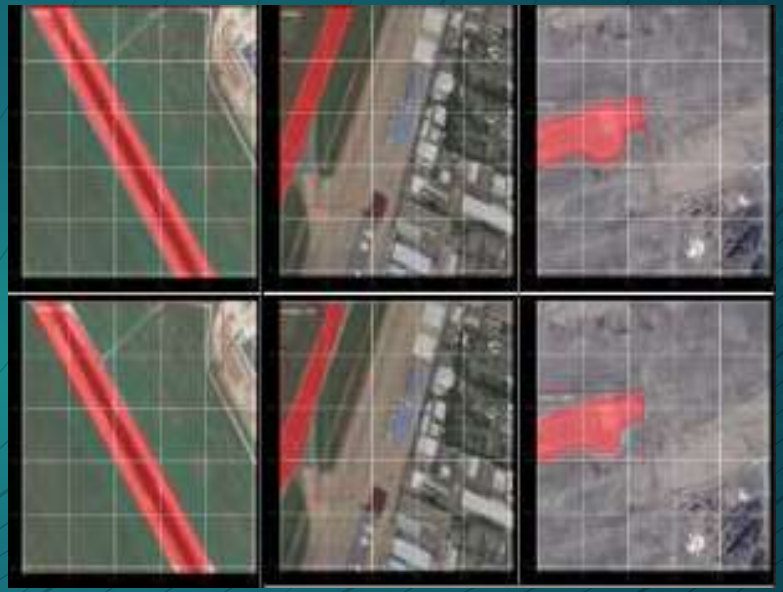
➢ Runway not detected :

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



➢ 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.



Mask R-CNN results on selected dataset. Above row shows true masks and lower row shows predicted masks.

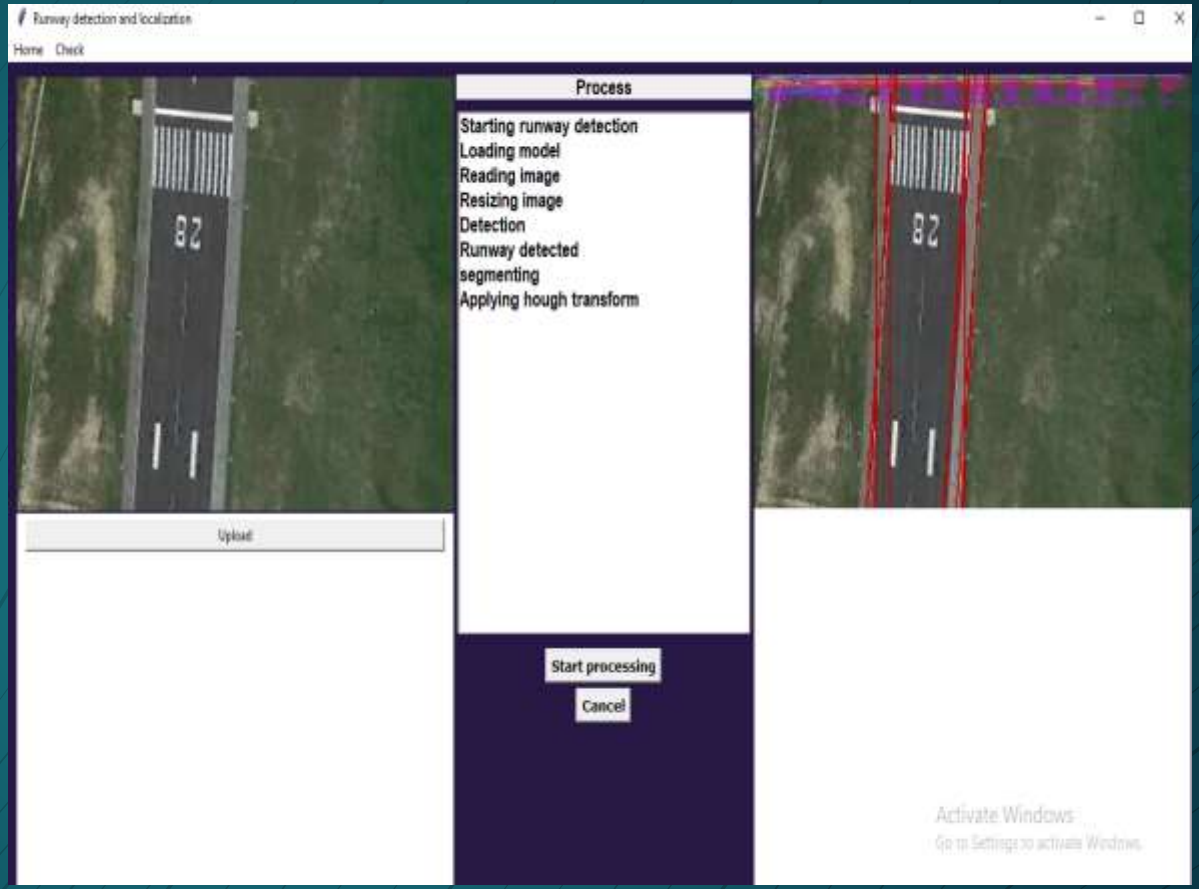
➢Hough Transform :

- 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.

Original image

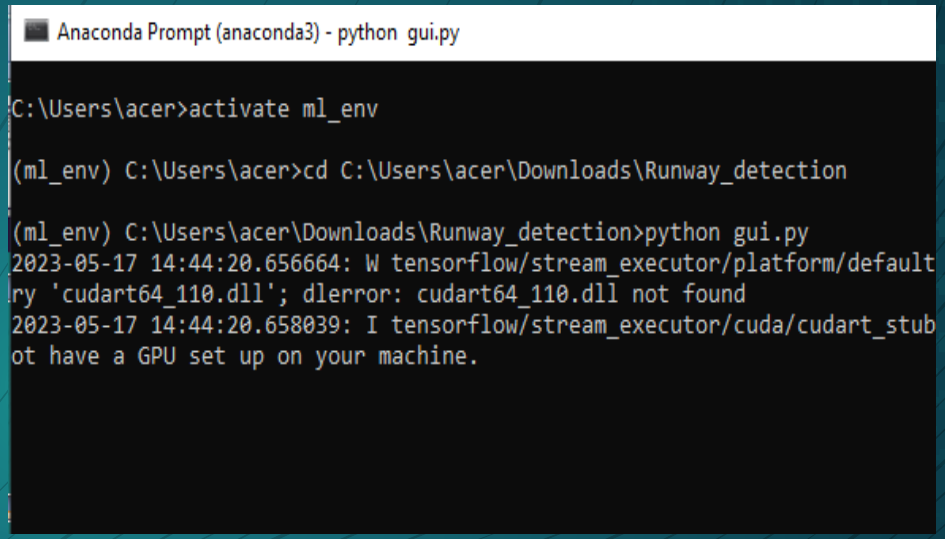
Result of applying HT

- 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.



WORKING

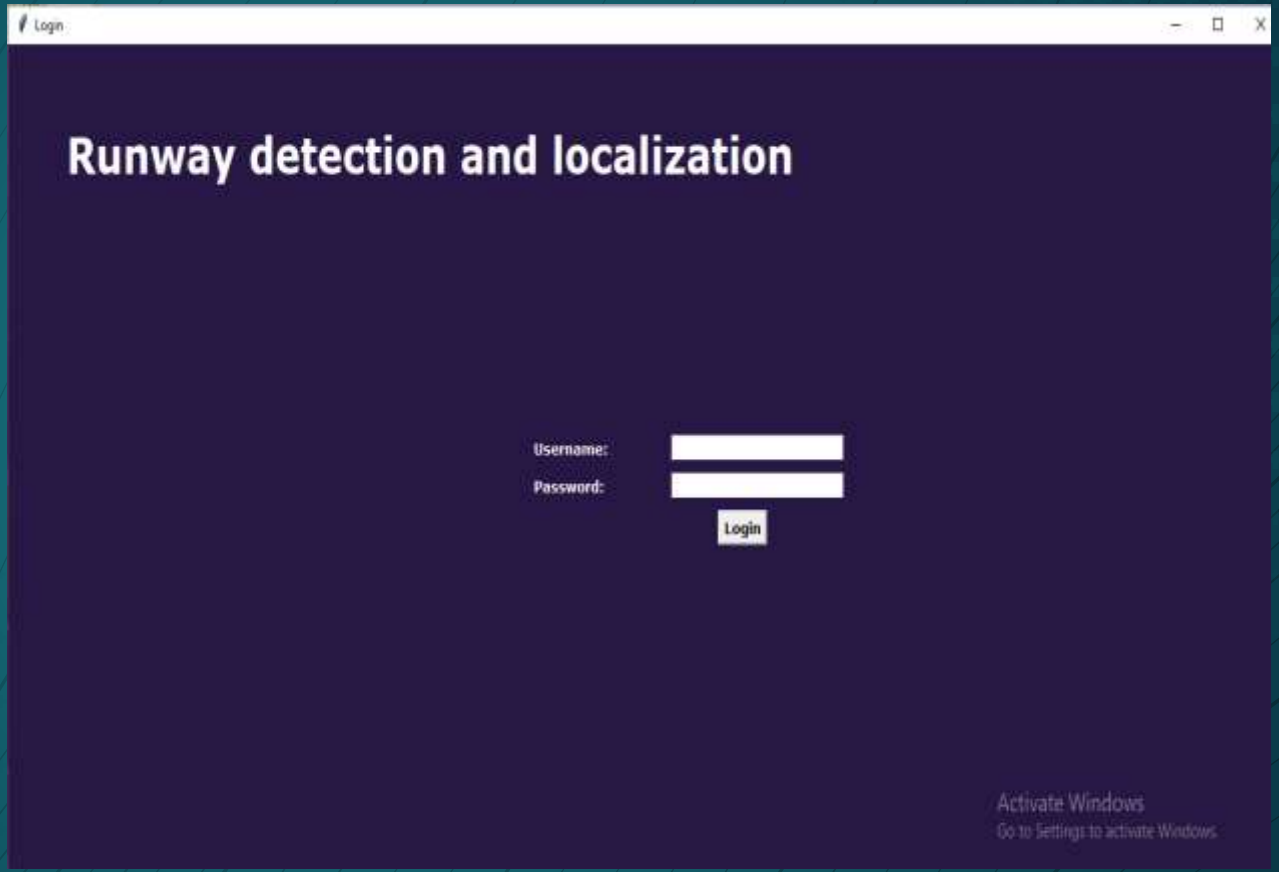
- It is basically a desktop application which is accessed through the Anaconda prompt .
- It is a build-in prototype model, where different sets of libraries are used which is installed in a virtual environment.
- When this environment gets activated only then the user can run anaconda prompt.
- It may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



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

1. Login page

- The user can access the login page by the username and password.
- The username and password is set in the code by the user.
- When the password typed is wrong then a alert message will be popped out showing wrong credentials.



2. Home Page

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

3. Check Button 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.

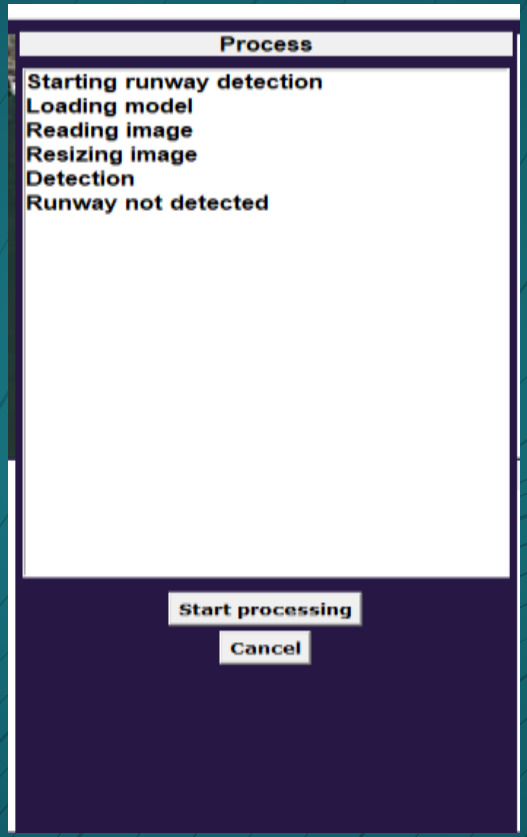
4. Upload Button Access

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

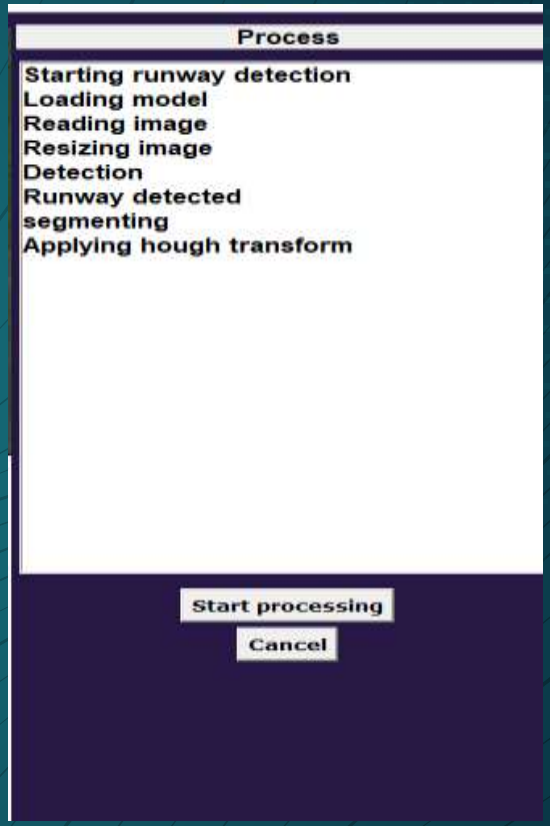
Two forms of aerial images from the dataset that where uploaded

5.Start processing

- After image is uploaded, then processing will be started
- Processing is of two forms mainly when in (a) the prediction without runway detection.
- In (b) the predictions when the runway is detected.



(a)



(b)

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 .

The runway detected and the localized image.

FEASIBILITY STUDY:

- Finds an accurate runway detection model.
- 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.

REQUIREMENTS:

HARDWARE REQUIREMENTS:

- Processor: i5 or i7
- RAM: 8GB (Minimum)
- Hard Disk: 500GB or above
- Mouse
- Keyboard

SOFTWARE REQUIREMENTS:

- Tool: Python IDLE
- Python: version3
- Operating System: Windows 10
- Front End: Python Tkinter

REFERENCES :

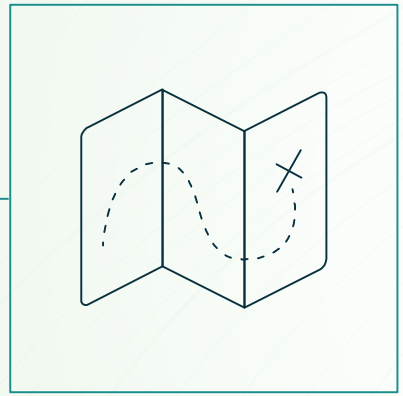
1. J. Akbar, M. Shahzad, M. I. Malik, A. Ul-Hasan and F. Shafait, "Runway Detection and Localization in Aerial Images using Deep Learning," 2019 Digital Image Computing: Techniques and Applications (DICTA), 2019, pp. 1-8.

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, “Airport Detection Base on Support Vector Machine from A Single Image,”2005 5th International Conference on Information Communications & Signal Processing, Bangkok, 2005, pp. 546-549.



Thanks!