A Deep Learning Approach For Airport Runway Detection and Localization From Satellite Imagery

Amine Khelifi[*,1], Mahmut Gemici[*,†,2], Giuseppina Carannante[*,3], Charles C. Johnson[‡,4] and Nidhal C. Bouaynaya[*,5]
[*]*Department of Electrical and Computer Engineering*, Rowan University, Glassboro, NJ
[†]*Intel Corporation*, Hillsboro, OR
[‡]*FAA Williams J. Hughes Technical Center*, Atlantic City, NJ
[1]amine2671993@gmail.com , [2]gemici34@rowan.edu, [3]carannang1@rowan.edu,
[4]charles.c.johnson@faa.gov, [5]bouaynaya@rowan.edu

***Abstract*—The US lacks a complete national database of private prior permission required airports due to insufficient federal requirements for regular updates. The initial data entry into the system is usually not refreshed by the Federal Aviation Administration (FAA) or local state Department of Transportation. However, outdated or inaccurate information poses risks to aviation safety. This paper suggests a deep learning (DL) approach using Google Earth satellite imagery to identify and locate airport landing sites. The study aims to demonstrate the potential of DL algorithms in processing satellite imagery and improve the precision of the FAA's runway database. We evaluate the performance of Faster Region-based Convolutional Neural Networks using advanced backbone architectures, namely Resnet101 and Resnet-X152, in the detection of airport runways. We incorporate negative samples, i.e., highways images, to enhance the performance of the model. Our simulations reveal that Resnet-X152 outperformed Resnet101 achieving a mean average precision of 76%.**

***Index Terms*—Machine learning; Object detection; Airport runway detection**

I. Introduction

The Federal Aviation Administration (FAA) maintains a comprehensive database of public and private landing zones that provide vital information to pilots in the event of an emergency. The accuracy of this database, which includes latitude and longitude coordinates, is of utmost importance to the FAA for ensuring aviation safety. Unfortunately, inconsistencies and a lack of reporting updates have led to inaccuracies in the database's landings and runways information. Such inaccuracies can have devastating consequences, as pilots may receive erroneous information, leading to fuel depletion or even fatal accidents. As such, it is imperative to validate the existing database and address any discrepancies.

Traditionally, the problem of detecting airport runways in aerial images has been predominantly approached using feature-engineering solutions and classical image processing techniques [1]–[4]. For instance, Han *et al.* [1] proposed an automated method based on identifying long rectangular shapes and using runway intensity and contrast for verification. Liu *et al.* [4] employed texture segmentation and shape detection in conjunction with the Hough Transform for runway segmentation. While these approaches have shown effectiveness, they are limited by manual feature engineering and may not capture the complexity of runway detection accurately.

Deep Learning (DL) algorithms offer an objective and data-driven approach to object detection and classification, making them a promising alternative to feature engineering, especially in the context of satellite imagery [5]–[7]. Pritt *et al.* [5] demonstrated the potential of DL to accomplish precise object detection in satellite imagery, utilizing an ensemble of convolutional neural networks (CNNs) and other Neural Networks (NNs). They classified objects and facilities from the Intelligence Advanced Research Projects Agency (IARPA) Functional Map of the World (fMoW) dataset. Given the advantages of DL techniques over traditional image processing approaches for detecting objects in satellite imagery, we propose an approach that leverages DL to accurately detect and locate airport runways.

In this paper, we present a comprehensive pipeline that utilizes DL techniques to detect and locate runways in satellite imagery, distinguishing them from highways. We incorporate image resolution manipulation as a critical parameter for identifying runway features and include negative samples, specifically highway images, in the training set to mitigate false detections. To evaluate the performance of our proposed method, we implement and compare two DL models based on the Faster Region-based CNN (R-CNN) architecture. Furthermore, we introduce a unique dataset specifically designed for runway detection in satellite imagery, which will be made publicly available to facilitate future research in DL-based object detection.

II. Proposed Methodology

This study outlines a comprehensive pipeline that utilizes DL to: create a high-quality labeled dataset of runway instances from satellite imagery, select suitable DL models, compute performance metrics for each model, identify areas for improvement, and propose a solution to improve model performance by including negative samples in the training set.

(a) A satellite runway image.

(b) A satellite image with two runway instances and their corresponding bounding boxes.

A. Data Acquisition and Preparation

To ensure the accuracy and reliability of the data source, the Airport Data and Information Portal (ADIP) provided by the FAA is utilized as the primary source of data. The ADIP can be accessed from the following link: https://adip.faa.gov/agis/public/#/airportSearch/advanced. This portal offers a comprehensive repository of latitude and longitude data for all accessible airports. We subsequently implement a manual filtering process to ensure that our dataset consists only of airports with paved runways, which are of interest in this study. Below, we provide a description of the data extraction process and outline the information required to replicate our labeling.

1) Google Static Maps API: To collect the satellite images required for this study, we developed a custom Python script that utilizes the Google static maps API, accessible at https://developers.google.com/maps/documentation/maps-static/overview. This API provides access to Google Earth's high-resolution imagery, allowing us to acquire images of our selected locations. By supplying the script with a CSV file containing the airport coordinates (latitude and longitude), the images are automatically retrieved and downloaded. Figure 1a illustrates a sample satellite image of runways obtained from Google Maps, providing a visual reference for the runway instances under investigation.

2) Image Annotation: We utilized the open-source graphical image annotation tool called "LabelMe" (available at https://pypi.org/project/labelme) to manually annotate airport runways in our dataset. The annotation process involves outlining bounding boxes and polygons around each runway. To ensure accuracy, we meticulously labeled each runway within multi-runway airports, resulting in a total of 4204 annotated runway instances out of 4000 images. Figure 1b showcases an example image with a labeled bounding box delineating the runways. These bounding boxes define the Region of Interest (RoI) for each image and serve as ground truth labels for the DL model to learn the corresponding coordinates.

B. Deep Learning Model

DL models have been extensively used for object detection tasks because of their exceptional performance [7]–[9]. However, there is still room for improvement by incorporating techniques such as transfer learning, hyper-parameter fine-tuning, and data augmentation, which can adapt the models to specific tasks and data characteristics. In our study, we leverage the Faster R-CNN (Region-based Convolutional Neural Network) model [10], and employ the aforementioned techniques to enhance the model's performance. The Faster R-CNN model offers several advantages, including end-to-end training, optimized computational efficiency, flexibility and scalability, and top performance in object detection tasks. These advantages make it a powerful tool for various applications requiring precise and efficient object detection. The Detectron2 framework, developed by the Facebook AI Research lab (FAIR), was utilized for implementing Faster R-CNN. Two distinct ResNet backbone architectures, namely ResNet-101 and ResNet-X152, were employed in our study. The models were sourced from the FAIR's model zoo, accessible at https://github.com/facebookresearch/detectron2/blob/main/MODEL_ZOO.md.

1) Multi-Scale Training: The performance of object detection models highly depends on the ability to accurately detect objects of varying sizes and scales [11]. This is particularly relevant in the context of aerial imagery, which often encompasses objects with significant size variations.

To ensure consistent image resolution throughout our dataset, we adopt a strategy of employing three different image sizes: 1500×1500, 3000×3000, and 6000×6000, while maintaining a fixed map size of 5000×5000 using the Google Maps static API. This approach ensures that the level of detail in the images remains uniform, allowing for equitable visibility of the underlying terrain in each image. We posit that the inclusion of images with diverse sizes and resolutions in the training set enhances the model's performance, enabling it to accurately detect objects across a spectrum of sizes and scales.

2) Data Augmentation: We used random brightness, contrast, saturation, lightning, rotation, flip, and grayscale transformation to augment the original dataset.

3) Transfer Learning: We imported the COCO [12]-trained weights and initialized the model with these pre-trained weights.

4) Fine-Tuning: Our optimal training configuration consists of a batch size of 16, a learning rate of 0.005, a momentum of 0.9, and a weight decay of 0.0001. The model was trained using stochastic gradient descent.

5) Incorporating Negative Samples: A significant challenge arises from the visual resemblance between runways and highways, as exemplified by a representative satellite image depicting a highway in Fig. 2. To address the concern of erroneous detection, we adopt a strategic approach in our training by incorporating negative sample instances that encompass highways. This incorporation of negative samples allows us to refine the model's discriminatory ability and enhance its accuracy in distinguishing between runways and highways. A carefully curated dataset comprising 1000 satellite images of highways was collected, encompassing diverse urban and rural landscapes. Stringent measures were taken to guarantee the absence of runways within these highway images, as the inclusion of runway instances in negative samples could potentially introduce ambiguity and hinder the model's ability to discriminate between runways and highways effectively.

C. Performance metrics and Evaluation:

The assessment of object detection models involves the utilization of metrics, such as precision, recall, and intersection over union (IoU) scores [13]. Precision quantifies the ratio of correctly detected objects to the total number of predicted objects, while recall measures the ratio of correctly identified objects to the total number of actual objects. The IoU score calculates the overlap between predicted and ground truth regions, indicating the level of agreement between them.

The Mean Average Precision (mAP) is a comprehensive metric that combines precision and recall by considering false positives (FP) and false negatives (FN) at a specified IoU threshold. For Faster-RCNN models, mAP is commonly used as the primary evaluation metric, measuring the average precision at a 0.5 IoU threshold across all object classes [10]. The 0.5 IoU threshold signifies a substantial overlap between predicted and ground truth bounding boxes. Evaluating the mAP provides a reliable and consistent measure of the model's performance across the entire dataset.

Fig. 2: A highway satellite image (rural view).

TABLE I: Model performance comparison

Backbone Architecture	Precision	Recall	F1-score	TP	FN	mAP
ResNet-101	0.99	0.82	0.9	81.8%	18.2%	73%
ResNet-X152	0.99	0.88	0.93	91.7%	8.3%	76%

1) Validation Data: To evaluate the performance of our models, we curated a validation set comprising a total of 300 images. Within this set, we carefully selected 200 images that contain 242 instances of runways. Additionally, we included 100 images that depict highways.

III. Results and Discussion

To ensure a fair comparison between the two models, we used identical configuration settings, including the learning rate, batch size, and the number of iterations, for both ResNet-101 and ResNet-X152 backbones. Despite achieving similar precision scores, ResNet-X152 outperforms ResNet-101 with a higher mAP score of 0.76. Table I summarizes the results. ResNet-X15 achieved a True Positive (TP) of 91.7% (vs. 81.8% for ResNet-101) and a False Negative (FN) of 8.3%. Figure 3 shows sample images of TP and FP predictions for both models.

The model's limited capability to detect runways in certain instances can be attributed to the suboptimal quality of the runway depicted in the satellite image. We show a sample image in Fig. 4. This is supported by the indistinctness of critical runway attributes, such as marking signs, which impairs the model's capacity to precisely recognize and localize the runway. These results highlight the importance of considering the quality of satellite imagery when training and evaluating

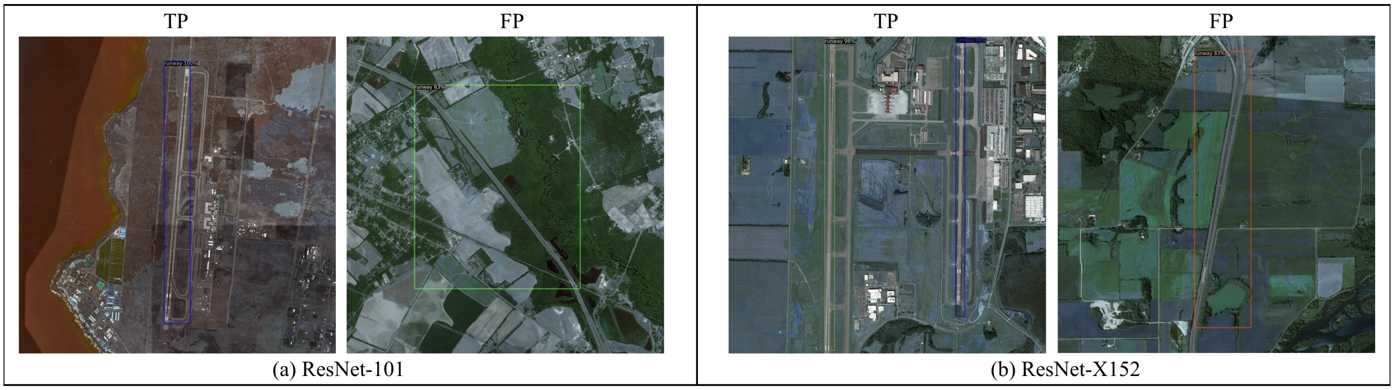

Fig. 3: Examples of True Positive (TP) and False Positive (FP) predictions for (a) ResNet-101 and (b) ResNet-X152 models.

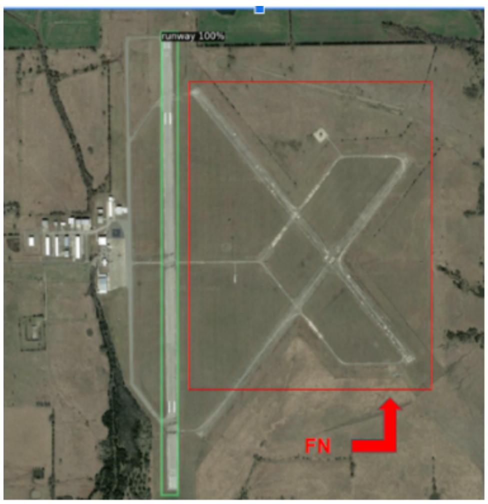

Fig. 4: Example of a false negative (FN) Detection.

runway detection models, as it can significantly impact their performance.

IV. Conclusion

This paper addressed the challenging task of runway identification and localization in satellite imagery using the Faster R-CNN region-based detector. Two Faster R-CNN models were evaluated and compared, trained, and validated on a meticulously labeled dataset of airports provided by the FAA. Various techniques were applied to enhance performance and prevent overfitting, including the incorporation of a balanced dataset of 1000 highway satellite images into the training set. The ResNet-X152 backbone achieved 99% precision and 88% recall. The findings have important implications for aviation safety, with the potential to update and enhance the FAA database for improved emergency landing location identification. Furthermore, the study contributes to the research community by creating a publicly available runway detection dataset (https://github.com/RowanMAVRC/Airport-High-Resolution-Satellite-Imagery-Dataset.)

Acknowledgment

This work was supported by the Federal Aviation Administration (FAA) contract No. 692M15-21-T-00001 and NSF ECCS 1903466.

References

[1] J. Han, L. Guo, and Y. Bao, "A Method of Automatic Finding Airport Runways in Aerial Images," in *2022 6th Int. Conf. Signal Process.*. IEEE, 2002, vol. 1, pp. 731–734.

[2] L. Wu and et al., "Canny Enhanced High-Resolution Neural Network for Satellite Image based Land Cover Classification and Its Application in Wireless Channel Simulations," *IEEE J. Selected Topics in Signal Process.*, 2022.

[3] R. Raja, S. Kumar, and M. Mahmood, "Color Object Detection based Image Retrieval using ROI Segmentation with Multi-Feature Method," *Wireless Pers. Commun.*, vol. 112, no. 1, pp. 169–192, 2020.

[4] Dehong Liu, Lihan He, and L. Carin, "Airport Detection in Large Aerial Optical Imagery," in *2004 IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2004, vol. 5, pp. V–761.

[5] M. Pritt and G. Chern, "Satellite Image Classification with Deep Learning," in *2017 IEEE Applied Imagery Pattern Recognit. workshop (AIPR)*. IEEE, 2017, pp. 1–7.

[6] Y. Zhong and et al., "SatCNN: Satellite Image Dataset Classification using Agile Convolutional Neural Networks," *Remote Sens. Lett.*, vol. 8, no. 2, pp. 136–145, 2017.

[7] S. P. Mohanty and *et al.*, "Deep Learning for Understanding Satellite Imagery: An Experimental Survey," *Frontiers in Artificial Intelligence*, vol. 3, pp. 534696, 2020.

[8] L. Fu, *Image Restoration Under Adverse Illumination for Various Applications*, Ph.D. thesis, University of South Carolina, 2022.

[9] X. Zhang, L. Han, L. Han, and L. Zhu, "How Well Do Deep Learning-Based Methods for Land Cover Classification and Object Detection Perform on High Resolution Remote Sensing Imagery?," *Remote Sens.*, vol. 12, no. 3, pp. 417, 2020.

[10] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Poposal Networks," *Advances in Neural Inf. Process. Syst.*, vol. 28, 2015.

[11] M. Khan and et al., "Deep Unified Model for Face Recognition Based on Convolution Neural Network and Edge Cmputing," *IEEE Access*, vol. 7, pp. 1440–1448, 2019.

[12] T. Lin and et al., "Microsoft COCO: Common Objects in Context," in *Proc. Eur. Conf. Comput. Vis.* Springer, 2014, pp. 740–755.

[13] M. Everingham, L. Van Gool, C. Williams, J. Winn, and A. Zisserman, "The Pascal Visual Object Classes (VOC) Challenge," *Int. J. Comput. Vision*, vol. 88, pp. 303–338, 2010.