CAIT-UTC-REG54

Rotorcraft Landing Sites Identification – Scaling and Generalization of the AI Model

FINAL REPORT

Submitted by:

Ghulam Rasool
Assistant Professor

Nidhal Bouaynaya
Professor

Nikolas Koutsoubis
Graduate Student

Abdullah Nasir
Graduate Student

Henry M. Rowan College of Engineering,
Rowan University
201 Mullica Hill Road, Glassboro, NJ 08028

External Project Manager
Cliff Charles Johnson
Rotorcraft, Unmanned Aircraft Systems (UAS), and eVTOL/Urban Air Mobility System Safety Section, ANG-E272 Aviation Research Division NextGen WJHTC Office
William J. Hughes Technical Center, Federal Aviation Administration (FAA)
Atlantic City International Airport, NJ 08405

In cooperation with
Rutgers, The State University of New Jersey
And
Federal Aviation Administration (FAA)
And
U.S. Department of Transportation
Federal Highway Administration

Disclaimer Statement

The contents of this report reflect the views of the authors, who are responsible for the facts and the accuracy of the information presented herein. This document is disseminated under the sponsorship of the Department of Transportation, University Transportation Centers Program, in the interest of information exchange. The U.S. Government assumes no liability for the contents or use thereof.

The Center for Advanced Infrastructure and Transportation (CAIT) is a Regional UTC Consortium led by Rutgers, The State University. Members of the consortium are Atlantic Cape Community College, Columbia University, Cornell University, New Jersey Institute of Technology, Polytechnic University of Puerto Rico, Princeton University, Rowan University, SUNY - Farmingdale State College, and SUNY - University at Buffalo. The Center is funded by the U.S. Department of Transportation.

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
CAIT-UTC-REG54		

4. Title and Subtitle	5. Report Date
Rotorcraft Landing Sites Identification – Scaling and Generalization of the AI Model	06 / 2022
	6. Performing Organization Code CAIT/Rowan
7. Author(s) Ghulam Rasool https://orcid.org/0000-0001-8551-0090 Nidhal Bouaynaya https://orcid.org/0000-0002-8833-8414 Abdullah Nasir https://orcid.org/0000-0001-6678-4846 Nikolas Koutsoubis: https://orcid.org/0000-0001-6195-9360	8. Performing Organization Report No. CAIT-UTC-REG54
9. Performing Organization Name and Address Henry M. Rowan College of Engineering, Rowan University, 201 Mullica Hill Road, Glassboro, NJ 08028	10. Work Unit No.
	11. Contract or Grant No. 69A3551847102
12. Sponsoring Agency Name and Address Center for Advanced Infrastructure and Transportation Rutgers, The State University of New Jersey 100 Brett Road Piscataway, NJ 08854	13. Type of Report and Period Covered Final Report 03/1/2021 – 02/28/2022
	14. Sponsoring Agency Code

15. Supplementary Notes
U.S. Department of Transportation/OST-R
1200 New Jersey Avenue, SE
Washington, DC 20590-0001

16. Abstract
The updated information about the location and type of landing sites is essential for the Federal Aviation Administration (FAA) and the Department of Transportation (DOT). However, the acquisition, verification, and regular updating of information about landing sites are not straightforward. The lack of current and correct information on helicopter landing sites is a risk factor in several accidents and rotorcraft incidents. The U.S. Helicopter Safety Team (USHST), of which the FAA is a key member, has identified and produced recommendations from their infrastructure working group to modernize and improve "the collection, dissemination, and accuracy of heliport/helipad landing sites" as a high priority to increase helicopter safety.
In the last couple of years, the Rowan team has been developing an AI-based system to identify landing sites from satellite images. The project activities were performed in collaboration with the FAA. The developed AI algorithm accepts latitude/longitude values and search radius (in miles) from the user and performs a detailed search for any landing sites, helipads, or landing ports. The results returned to the user consist of satellite images marked with possible landing sites and corresponding latitude/longitude coordinates of the identified landing sites. The AI algorithm can also scan whole cities, towns, or extensive areas to locate and mark landing sites. The team has updated the FAA's 5010 databases of helipads, heliports, and landing sites using the developed AI.

17. Key Words	18. Distribution Statement
Helipad, detection, machine learning, deep neural networks, aviation safety	

19. Security Classification (of this report)	20. Security Classification (of this page)	21. No. of Pages	22. Price
Unclassified	Unclassified	36	

Acknowledgement

The project team would like to express gratitude to the National University Transportation Center Consortium led by Rutgers' Center for Advanced Infrastructure and Transportation (CAIT) for their generosity for this research project. The authors would also acknowledge the support provided by Cliff Charles Johnson, the external project manager and stakeholder from William J. Hughes Technical Center, Federal Aviation Administration (FAA). Cliff was instrumental in the project's conception and provided multiple helipad datasets used for the training of machine learning algorithms. He regularly met with the team and guided the project at every stage.

We would also like to thank LZControl for their guidance and assistance with this effort. Via a Cooperative Research and Development Agreement with the FAA, LZControl provided a set of data from their system and subject matter expertise which provides landing zones for helicopters across the U.S., often complementing the FAA's 5010 database and including locations/sites not present in the FAA's 5010 system.

Table of Contents

1 Introduction

1.1 Background and Motivation

Accurate information about the location and type of rotorcraft landing sites is an essential asset for the Federal Aviation Administration (FAA) and the Department of Transportation (DOT). However, the acquisition, verification, and regular updating of information about these landing sites is a challenging task. The lack of reliable information on helipad sites is a risk factor in several accidents and incidents involving rotorcrafts. The U.S. Helicopter Safety Team (USHST), of which the FAA is a key member, has identified and produced recommendations from their infrastructure working group to modernize and improve "the collection, dissemination, and accuracy of heliport/helipad landing sites" as a high priority to increase helicopter safety.

There are thousands of landing locations for helicopters spread across the United States. In general, rotorcraft operators can get information about helipads, heliports, and landing sites using various databases, such as the FAA's 5010 database. However, it is also well-known that the 5010 database and similar databases contain multiple inaccuracies where some helipads in the database may no longer exist or their coordinates are imprecise, and other helipads are missing from the database. The unreliability of this database is a consequence of the fact that there is no system to verify that coordinates remain accurate, nor is there a system to search for unreported helipads.

In this project, we propose a machine learning solution to identify helipads, heliports, and other landing sites, from aerial imagery using convolution neural networks or CNNs. We built a comprehensive database by manually checking the FAA and other databases with satellite images from Google Earth. We subsequently trained and validated different state-of-the-art CNN models to determine an appropriate machine learning model for this task.

The proposed machine learning solution based on modern artificial intelligence (AI) techniques will allow the FAA and USDOT to automatically maintain an updated database of helipads, heliports, and landing site infrastructure for the rotorcraft community. This work presents the first step towards autonomous identification of specialized heliport infrastructure and can be optimized with minimal cost using Google Earth API. The results of this project will help the FAA and USDOT achieve the first strategic goal of "Improving durability and extending the life of infrastructure" by providing an updated record of the infrastructure without committing additional resources for data collection and recording.

1.2 Related Work

We can group the literature of identifying helipads from satellite or aerial imagery into two main approaches. The first is a model-based approach, which relies on domain expert knowledge to ex-

tract features that can be used to identify helipads from images. A common feature used to identify helipads is the "H" marking [Prakash and Saravanan, 2016, Patruno et al., 2017]. For vision-based autonomous landing systems, an improved version of the Scale Invariant Feature Transform (SIFT), called Speeded Up Robust Features (SURF), was used in [Prakash and Saravanan, 2016] to perform feature points matching and tracking. Features points are compared to points in an "H" template to determine the similarity of the template and the aerial image.

In [Rungta et al., 2020], the detection process consists of finding candidate helipads based on the following four properties: (1) a bold circle surrounding the "H", (2) presence of "H" in a bright color inside this circle against a dark background, (3) "H" is centered at the center of the circle, and (4) intersection of diagonals of "H" at the center of the circle. A Hough transforms was used to identify circles[Rungta et al., 2020]. Due to a large number of false positives, the authors used three tests to eliminate these false positives. None of these tests are precise and as a consequence, error ranges were added based on experiments. After a helipad has been detected, a Median Flow tracker [Zdenek Kalal, Krystian Mikolajczyk, and Jiri Matas, 2010] was used to track the region.

A vision-based helipad detection algorithm based on curvature was proposed by Patruno et al. [Patruno et al., 2017]. The method creates blobs of connected pixels, and exploits some intrinsic properties of each blob, such as the location of its center of mass, the Euler number, the eccentricity, the perimeter, and the area, to identify the blobs which represent the helipad marks, namely the character "H" and the circumscribing circles. The Euler number is an integer value defined as the number of connected components minus the whole number. In particular, the Euler number is equal to zero for circle blobs and one for "H" blobs. A final classification level checks the ratios between the areas and perimeters of blobs against expected values. Following detection, an identification step checks if the Euclidean distance of the centroids of the detected blobs and the ratios of related areas and perimeters are still met [Patruno et al., 2017]. Once the helipad marks have been identified, the Canny edge detector is performed in order to extract the 12 corners of "H" edge. Instead of using feature extraction operators, such as the Hough transform and line following algorithms, the authors used a radius of curvature for every 2-D point of "H" edge to detect the corners of interest. A big radius value denotes that the point is far from a corner while a small value indicates that the point might be a possible candidate to be a corner. Three checks are performed for all the possible corner candidates, based on the knowledge of "H" size and exploiting the Euclidean distances between these points and the centroid of "H" contour.

Although quite exhaustive, these model-based detection algorithms have many restrictions. First, they were shown to work only in simple simulated environments and may fail in more complex environments. Secondly, these algorithms have limited effectiveness at further distances and angles. Some of these issues were addressed in [Pierre et al., 2018], where the authors mainly relied on flat ellipse detection as it is the most visible feature of a helipad seen from long distances.

An adaption of the Hough transform was devised for the specific case of very flat ellipses. A validation step using many other properties and visual clues performs the verification of the presence of the helicopter landing platform in the research areas delimited by the obtained ellipses.

The main advantage of the model-based approach is its explainability and its relatively good performance on small datasets with no prior labeling. However, while model-based methods can identify helipads that adhere to the recommended standard set in the FAA's 150/5390-2C, neither the circle nor the "H" is required for building a helipad. Model-based methods will need to consider all possible features of all types of helipads/heliports, including those that do not adhere to the recommended standard, to generalize their performance [FAA, 2012].

Data-driven algorithms, on the other hand, involve the collection of large amounts of labeled data, autonomously learning salient features from the raw data, and identifying helipads based on learned features. As such, data-driven systems can identify complex patterns of helipads that may be hard to model. The price paid is the large data and computational resource requirements. To the best of our knowledge, data-driven approaches to identify helipads are under-explored, despite the growing prevalence of learning systems in real-world applications. Nonetheless, there are online systems available.

HelloPad is a system that uses a machine learning algorithm to identify helipads within a specified region [Walker, 2019]. The system uses a sliding window and a trained neural network model (ResNet) to identify if a helipad exists at a given location. HelloPad reported 67.2% precision and 90% recall in a Los Angeles downtown area. However, HelloPad collected negative (non-helipad) examples from urban settings, and will likely not transfer well to all areas of the U.S.

Add David's work here.

2 Deep Learning Methods

2.1 Classification Using Convolutional Neural Networks (CNNs)

Object detection and identification requires considerable domain expertise to design features that transform the raw data (such as the pixel values of an image) into a lower-dimensional representation that is discriminatory for the input. Convolutional Neural Networks (CNNs) are designed to process multidimensional data arrays, such as images, by automatically discovering the representations needed for detection or classification. There are three types of layers in a CNN: convolutional layers, pooling layers, and fully connected layers. Each convolutional layer obtains, through convolutions followed by non-linear operators, representations that are important for the classification task. A hierarchical composition of these representations (starting with the raw input), where each representation is fed to the next convolutional layer, leads to learned features that are optimal for

discrimination. The first (convolutional) layers typically learn low-level features, such as edges, and later layers extract more complex semantic features. The key aspect of CNNs is that these layers of features are not designed by human engineers or domain experts: they are learned from data [LeCun et al., 2015].

A problem with the output feature maps is that they are sensitive to the precise location of the features in the input. This means that small variations in the position of the feature in the input image will result in a different feature map. One approach to address this sensitivity is to coarse-grain the position of each feature through down-sampling, referred to as "local translation invariance". The role of pooling layers is to summarize the feature maps by down-sampling, i.e., discarding the finer details that may not be useful to the task, creating an invariance to small shifts, while maintaining important structural elements. A typical pooling unit computes the maximum value for each patch of the feature map.

Layers of convolutions, non-linearities, and pooling are stacked to learn robust optimal features for the data, followed by fully-connected layers that form the classifier for the extracted features. Backpropagating gradients through a CNN is as simple as through a regular neural network, allowing all the weights in all the filters to be trained.

2.2 Interpreting and Explaining the Predictions of CNNs

While CNNs have achieved higher-than-human accuracy in many computer vision tasks, they provide little insight into computations that they perform to make these decisions or predictions. With the composition of convolutions, non-lineariries, pooling and fully-connected layers, very complex functions can be learned, making deep learning models black boxes. This poor interpretability significantly hinders the robustness evaluation of the network, its further optimization, as well as understanding the network adaptability and transferability to different datasets. In the case of helipad detection, this question becomes "Does the network detect salient features of helipads in the image, or does it detect other features that typically correlate with the presence of a helipad?". An understanding of the learning process will allow for the identification of cases where the algorithm might fail, and also build trust in learning systems to allow for their safe deployment.

An intuitive approach to understand the inner workings of deep learning models (such as CNNs) is the *gradient saliency map*. This approach computes the gradient of the class score with respect to the input image; thus, highlighting the areas of the input image that are discriminative with respect to the predicted class [Simonyan et al., 2014]. A popular gradient saliency method is the Gradient-weighted Class Activation Mapping (Grad-CAM). Grad-CAM uses the gradient information flowing into the last convolutional layer of the CNN to assign importance values to each neuron for a particular decision of interest [Selvaraju et al., 2019].

In this project, Grad-CAM provides a multifaceted advantage. First, the saliency map will be able to verify that the network classifies imagery as helipads because of the presence of helipads and not supporting facilities. Second, it can help with understanding and mitigating false positives, i.e., the non-helipad samples classified as a helipad. Lastly, the saliency map can help locate the helipad, which will allow for larger regions to be searched for helipads.

2.3 Detection Using Deep Neural Networks

2.3.1 You Only Look Once (YOLO) CNN architecture

The model designed in this work utilized the YOLO model architecture. This is one of the most popular models used for object detection, and is known for its high accuracy and fast computation speeds. This model i based on the idea that a single network predicts bounding boxes and class probabilities directly from full images in one pass. This allows for end to end optimization specifically on detection performance.

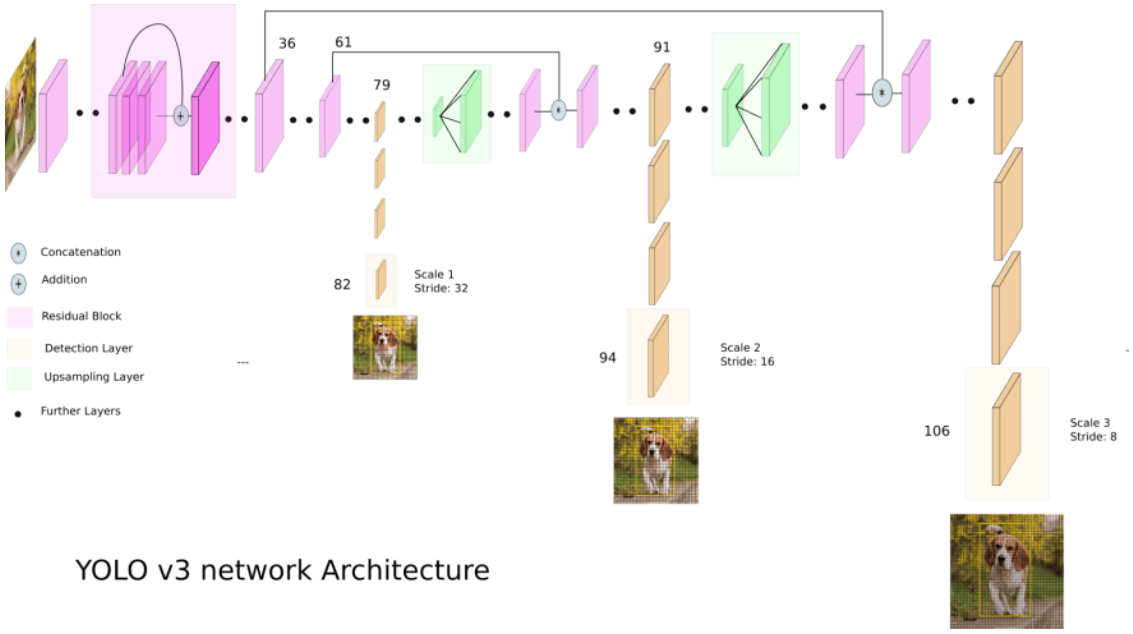

Figure 1: Model Architecture of YOLOv3

Figure 1 displays the YOLO v3 model architecture. The input is a batch of imagery data, in our case it will be places of interest with the potential to contain helipads. The output is a list of bounding boxes along with recognized classes. Then utilizing intersection over union, or the overlap of the predicted bounding box compared to the ground truth labels, a confidence score can be given on the accuracy of the detection. The model is 53 layers deep and pretrained on the imagenet model.

2.3.2 SSD MobileNet Architecture

The SSD architecture is a single convolution network that learns to detect objects in one pass. The SSD network consists of base architecture (MobileNet in this case) followed by several convolution layers. SSD architecture allows faster processing compared to other models with a tradeoff for accuracy.

2.3.3 Fast/Faster R-CNN

Fast/Faster R-CNN are also very popular convolution architectures,Fast R-CNN uses selective search to generate region proposals.However, Faster R-CNN uses region proposal network (RPN) for generating region proposals and a network using these proposals to detect objects. The time cost of generating region proposals is much smaller in RPN than selective search. RPN ranks region boxes and proposes the ones most likely containing objects.

2.4 Model Training, Testing and Validation Details

We trained 3 different models YOLO, SSD Tf and Pytorch Detectron to collect sample results for comparison. Model comparison allowed us to compute different trade offs such as precesion and accuracy. Based on our sample set we deemed YOLO V3 to be most appropriate for our project. Following are model comparison metrics.

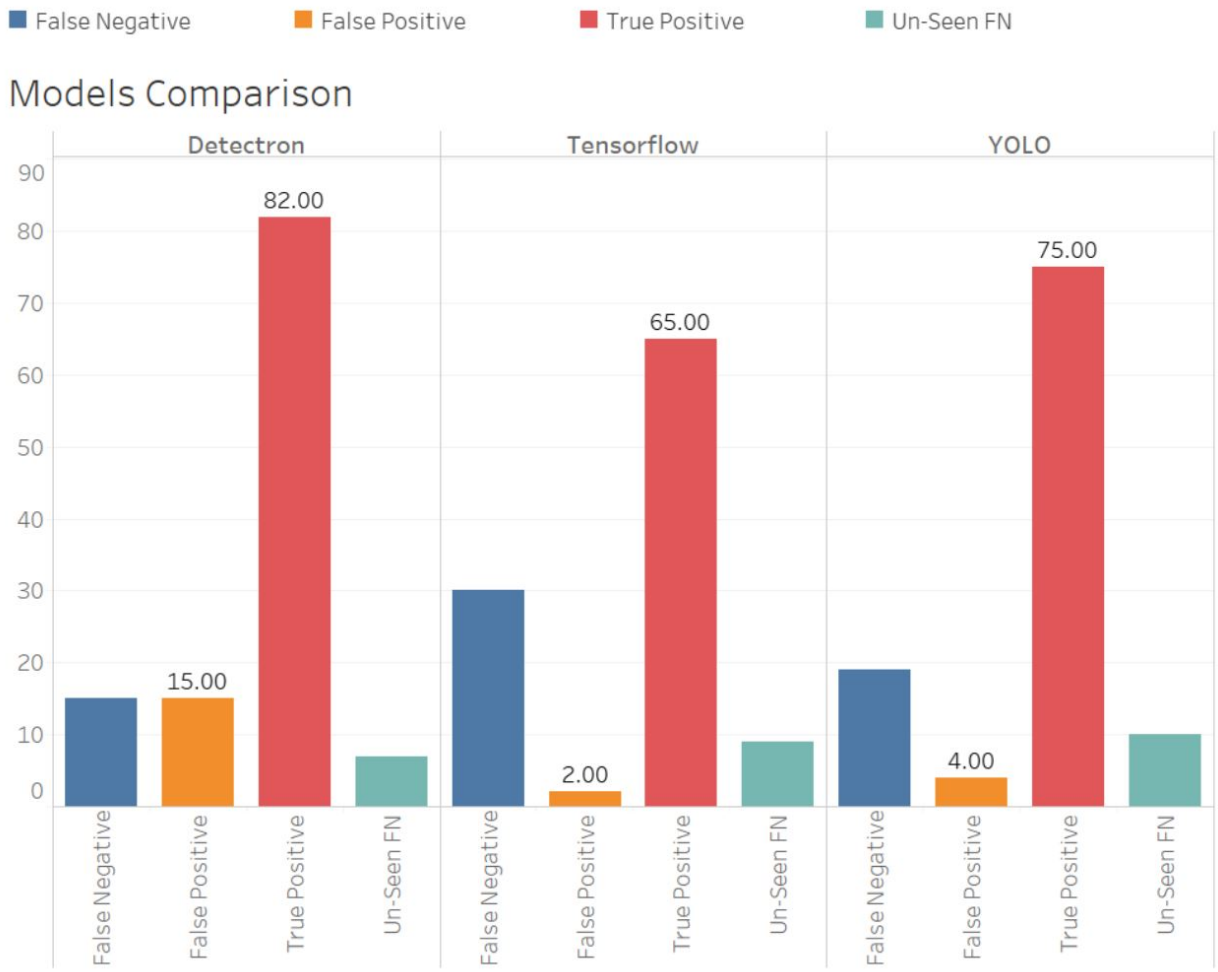

Figure 2: Model Comparison

Figure 2 compares different performance metrics, YOLO showed low number of False Positives and good True Positive Numbers.

Our Model showed promising performance numbers as it had a 86% True Positive and 14% False Positive on validation set.

3 Rotorcraft Landing Site Dataset

We acquired three datasets through the FAA, one dataset from the Iowa DOT website and one dataset from ArcGIS. These five datasets provide the longitude and latitude of potential helipad landing locations. We used Google Earth's API to extract the corresponding images as well as to sample negative helipad locations. We noticed some discrepancies in the FAA datasets and manually curated the coordinates to ensure accuracy for our use cases. In the following, we will

Figure 3: Sampling images from Google Earth for building datasets of positive (helipad is present) and negative examples (no helipad is present). The dataset will be used to train machine learning and AI models. The sampled area is enclosed in the black box. There is no helipad inside the sampled area (negative example). However, a helipad is present just outside the sampled area. The area containing helipad can be also be sampled as a positive example.

elaborate on each dataset, data cleaning approach, and our method for collecting negative samples (satellite images with no designated helipads or landing sites present). Figure 3 shows the sampling process for image collection for positive examples (a helipad is present in the satellite image) and negative examples (helipad or landing site is not present in the image).

3.1 Google Static Maps API

We used Google static maps API to collect satellite imagery of positive (helipad) and negative (non-helipad) locations. The service is accessed by sending an *HTTP* request with a query containing the desired parameters. The Google server responds with an image based on the provided parameters. The parameters used here are: center, zoom, the size, and maptype. The center provides the coordinates of the center of the image. Zoom determines the zoom level, which defines the resolution of the current view. Size determines the number of pixels in the image. Maptype determines the type of image to be retrieved (as Google maps contains road maps). For the purposes of this project, size was set to the maximum value of 640×640, and the maptype was always satellite. The center was set to the desired coordinates to be sampled for the image. The highest resolution images are available at a zoom of 20; however, a zoom of 18 was used instead. At zoom 20, some images did

not include the designated helipads as shown in Figure 3. The difference between the two zoom levels can be seen in Figure 4. A lower zoom results in a larger area that will allow for sampling helipads using fewer API calls. There is a cost associated with making API calls beyond a certain limit, so efficiency of calls becomes important for scaling up the model to large areas.

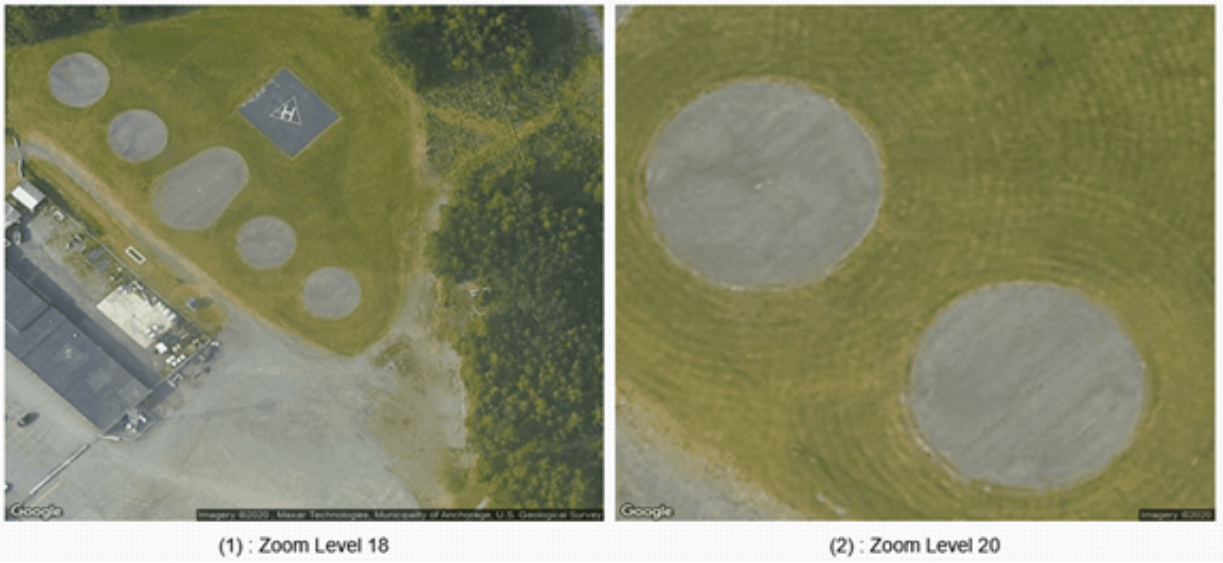

Figure 4: Zoom level in Google Map API. Left image is downloaded at the zoom level of 18 and the right image at the zoom level of 20. At a zoom of 18, numerous possible parking pads and a helipad are visible. The helipad is not visible at the zoom of 20.

One major issue with the initial databases is the incorrect reporting of landing site coordinates. In our experiments, a coordinate was considered correct if there was a landing area present in the Google imagery taken of the area. This is to allow for a margin of error in the reported coordinates. The margin is considered acceptable as it is believed to be reasonable for a pilot to identify a helipad within the given area. However, there are few cases where helipads would be within a reasonable range of the coordinates, yet not present within the imagery being sampled. Figure 3, shows a case where there is a helipad near the coordinates, however the helipad is outside the range that was annotated. A lower zoom could be used to sample a larger area, however while this may still be within an acceptable margin of error, the markings on helipads become less noticeable.

Another known issue is the recency of Google's satellite imagery. The images used in Google maps are not real-time images, but rather imagery taken during an area survey. This means that the overhead view that was sampled does not actually reflect the current state of the area. Google attempts to keep the images up to date such that the available imagery should be less than three years old; yet this may still lead to inaccuracies in landing site locations

Google Maps maintain a database that covers most of the world; however, it does not contain

high resolution imagery for every coordinate in the world. Typically, at higher levels of zoom, there are fewer coordinates with available imagery. Even when using a zoom of 18, there are a few coordinates that simply did not have the imagery available. If a zoom of 20 were used, there would likely be fewer locations from where images could be downloaded.

3.2 Building the Dataset

3.2.1 Positive Examples

Areas with helipads are needed to create a positive dataset for the training of the machine learning model. While areas can be randomly sampled using Google Maps API and helipads in those areas labeled, this would be an incredibly inefficient process. There is an extremely low probability that a randomly sampled location would contain a helipad. We used the initial FAA, IOWA DOT, and ArcGIS helipad datasets to sample positive areas, The FAA's 5010 was the largest database. To ensure accuracy, all coordinates were manually annotated so that only coordinates where a helipad would be visible in the collected image was added to the training set. From an initial $6,333$ coordinates in the dataset, only $3,887$ were manually annotated to be helipads. An additional 157 positive coordinates were added from other databases provided by the FAA, including the Lifeflight of Maine dataset

Two publicly available datasets were used. The first is a dataset found on ArcGIS containing the coordinates of hospital helipads found in California. This dataset contained 170 coordinates, and after annotation, 169 of these coordinates were used. The second is Iowa DOT's dataset, which listed 126 locations, and 111 of these coordinates were considered to contain helipads.

3.2.2 Negative Examples

A negative (non-helipad) set of images is also needed to train the machine learning model. The negative set was collected using random sampling of Google Maps. These random samples were manually checked to ensure that they did not contain any landing site. As the current goal is to identify helipads in the U.S., the sampling was limited to an area such that the sampling region includes most of the mainland U.S. However, most of these samples were of forested areas and farmlands and contained very few urban areas. This could bias the network to predict helipads mainly in urban areas It is therefore important to sample negative locations from urban areas as well. It is noted that urban areas will likely have a higher helipad density, and thus a helipad will be more likely to be found there. To lessen this risk, locations like Washington D.C. and New York City were chosen due to the lower density of helipads. In New York City, ownership of rooftop helipads became more restricted after the 1977 crash at the Pan Am building, along with noise

complaints continuing to restrict helicopter flights. Washington D.C. is in restricted airspace and allows only a few helipads to operate.

3.3 Benchmark Dataset for Training CNNs

After careful data collection, labeling, and organization, a helipad identification benchmark dataset was created. The positive set contains $4,324$ samples. Some areas are more represented than others, as some of the datasets used were specific to certain regions. However, the largest dataset making up over 80% of the final dataset is the FAA's dataset spread over the United States and its territories covering different types of landing areas, including helicopter parking pads, helidecks, Emergency Helicopter Landing Facilities (EHLFs), and heliports.

The negative set was created by randomly sampling 5,000 coordinates. A total of 2,000 of these coordinates were from the mainland United States and contained woodland and other rural areas. The remaining 3,000 negative images were sampled from urban areas, such as San Jose, Washington D.C., New York City, and San Antonio.

Figure 5: Sample aerial images from the dataset that we developed as a part of the project. While the term helipad is used for the positive set, the dataset also contains areas that helicopters are intended to land at, e.g., helicopter runways.

The benchmark dataset has $9,324$ satellite images labeled as either helipad or non-helipad. Figure 5 shows some of the images in the dataset. On the left, we show some landing locations, including helistops, helidecks, and helicopter runways. On the right, we present some randomly sampled imagery including rural and urban areas. It is noteworthy to mention the variety of landing sites shown in Figure 5. In particular helipads have different sizes, as their minimum required lengths are decided by the rotor diameter of helicopters intended to land. This causes the areas they

represent in squared meters to be different. Other factors, such as the zoom level which takes into account the distance from the satellite, the elevation, and the latitude add to the complexity of the landing sites imagery.

3.4 Expanded Database Construction

With the initial database constructed, further work was conducted to expand this database, focusing on particular places of interest (POI). The primary POI's Searched for in this endeavor were hospitals and Airports as they are two very promising locations to contain helipads. These helipads also may vary in their shape and design, giving the models better training data to further improve its ability to generalize on helipads. Utilizing the Statics map API To query these places of interest A script was designs to query and collect images for 825 cities from throughout the United States. This search returned 15000 places of interest from throughout the country that could be used to simultaneously train the model to detect a wide range of helipad types, and expand the current data set further. After checking this data and removing any duplicate imagery, the data set had 9089 hospitals or airports that were ready to be searched for helipads. This data set is titled Helipad_data set. The labelling process involved Drawing ground truth boxes around helipads within this dataset. The Model would then compare its predictions to the ground truth and use intersection over union to determine the confidence level of a given prediction.

Figure 6: Examples of Hospitals in the dataset with Helipads

Figure 7: Examples of Airports in the dataset with Helipads

Take out the sections on David's work related to LA area.

4 Helipad Search in Large Areas

Using the CNN model validated in the previous section, it is now possible to identify imagery with helipads, which can be used to verify the accuracy of coordinates in helipad databases. This system can then be extended to be able to detect helipads within a designated region. In computer vision, the distinction between identification and detection is that identification can determine the presence of an object, while detection determines where in the image an object is. In this section, we extend the problem of helipad identification from aerial images to the detection of helipads from a larger area, e.g., downtown Los Angeles. To solve this new problem, without requiring new labeling, we use a sliding window approach to determine where in a larger image a helipad is.

4.1 Searching for Helipads in Large Areas

Sampling a larger Google Earth area can be done using a lower value zoom, dividing it into sections, then upsampling the images. This approach would minimize the number of API calls; However, the images retrieved will be of lower resolution. The second approach would be to sample using a higher zoom for higher resolution imagery, then combine the samples to form a larger image referred to as a *collage*. This collage can then be searched for helipads with an overlapping sliding

window. A mapping between the latitude/longitude coordinates and pixel values must be derived. Google has provided the following relationship:

$$\frac{\text{meter}}{\text{pixel}} = 156543.03392 \times \frac{\cos(\text{latitude} \times \frac{\pi}{180})}{2^{\text{zoom}}} \tag{1}$$

The distance represented by a pixel decreases as we sample further from the equator. Equation (1) does not factor in elevation, and may cause issues at different elevations.

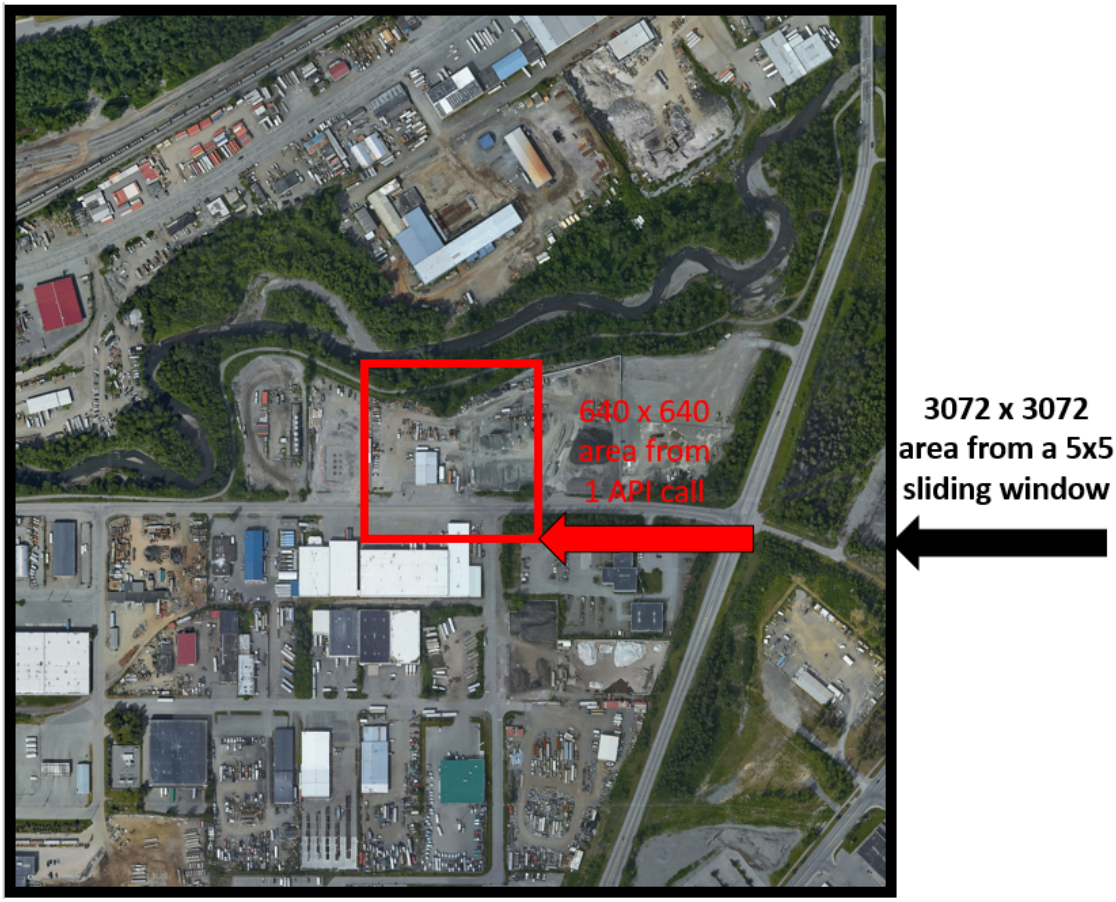

Figure 8: Searching helipads in large ares. The area scanned in a 5×5 collage settings vs. a single API call.

Assuming that the circumference of the earth is 40.075 million meters and taking elevation into account, we can derive the following mapping from pixels to change in latitude/longitude.

$$\frac{\Delta \text{ latitude}}{\text{pixel}} = 156543.03392 \times \frac{\cos(\text{latitude} \times \frac{\pi}{180})}{2^{\text{zoom}} \times 111320} \tag{2}$$

$$\frac{\Delta \text{ longitude}}{\text{pixel}} = 156543.03392 \times \frac{1}{2^{\text{zoom}} \times 111320} \tag{3}$$

An example of the created collage can be seen in Figure 8. This collage is created from a 5×5 sliding window, and shows an area about 25 times larger than the initial aerial images, while still keeping the level of detail at a higher zoom. Sub-images can then be extracted from this area to search for helipads.

Figure 9: Los Angeles (LA) region to be sampled. As the network was not trained on a similar cityscape, the top half of the area was used to supplement the training dataset and the model was tested on the lower half.

5 Helicheck Web App

With the ability to detect Helipads in mass, The Next step was to be able to directly search if a helipad existed in a specific place of interest in a quick and accessible way. This lead to the design of the Helicheck Web applications that provided a place that one could search a set of coordinates and return and image of the location ran through the Yolo model to detect a helipad, if a detection is made, a bounding box would be drawn around the helipad with an associated confidence score that the detection is accurate. The coordinates for most hospitals can be found wit a quick search, but information of the availability of a helipad is not such a trivial find, this webapp allows for the quick check to determine if a given place of interest has a helipad available

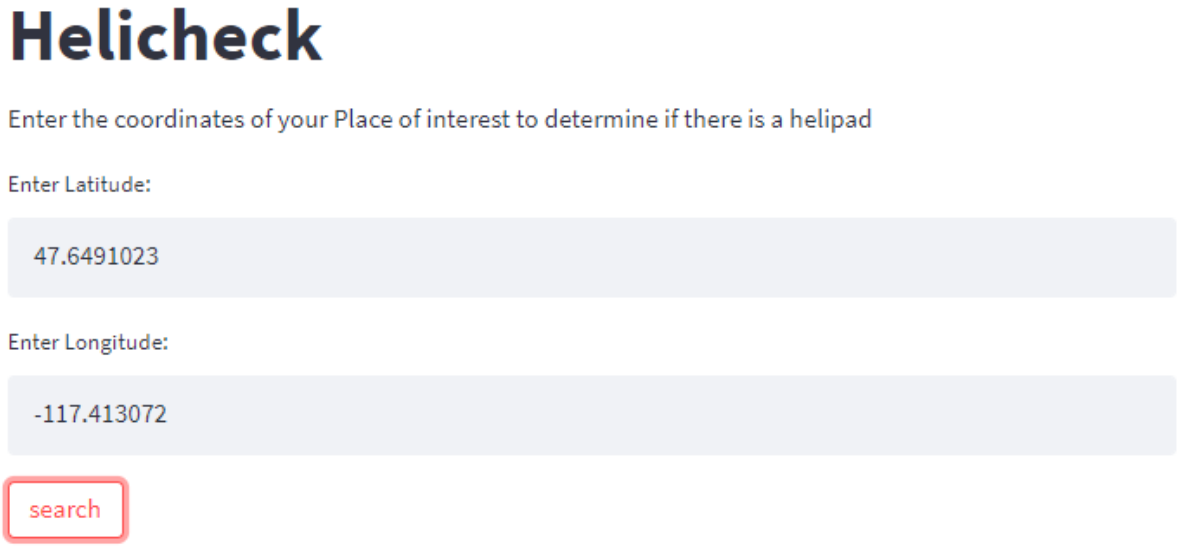

Figure 11: Examples of Airports in the dataset with Helipads

Figure 11 displays the front end design of the Helicheck webapp. the app was designed to be simple and intuitive, simply add the latitude and longitude of the location you wish to check for a helipad

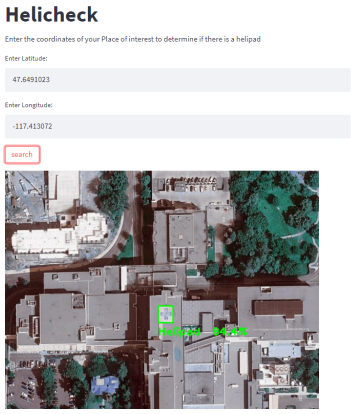

Figure 12: Example of Helicheck web app detection

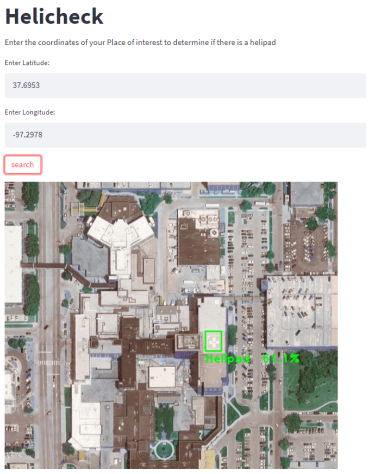

Figure 13: Example of Helicheck web app detection

Figures 12 and 13 display the web app functioning. these user simply inputs the latitude and longitude of the place they would like to query and return an image with the helipad drawn in a bounding box with the models confidence of detection.

6 Database Development

With a large data pool of places of interest with the potential to contain helipads, in addition to searching a single area at once, it is also useful to contain a table of all the locations with a field denoting if a helipad was detected in that location. This lead to the design of the csv confirmation script. This framework also takes a csv file of locations with name latitude and longitude as opposed

to the prior designs that took imagery data. Th csv Confirm script then queries these images from the google maps API and runs them through the model. IT then outputs 2 csv files, once containing all the locations with name, lat, and lon, that had a detection. The second csv returns the same list input into the script, however there is now an additional field called detection with confirmed or unconfirmed on each element. using this file one can quickly determine which locations that they seek to identify have helipads

name	lat	lon	detection
(Temporary Location) Greater Danbury Community Health Center - Vaccine Clinic	41.39202	-73.4509	unconfirmed
10327 East Highway 120	37.6661	-120.993	unconfirmed
127 S San Vincente	34.05221	-118.244	unconfirmed
1414 Cascade Medical Center	47.62319	-122.185	unconfirmed
1423 Magnolia Avenue Care Center	39.74045	-121.85	unconfirmed
1901 Swhk Dodgen Loop	31.07255	-97.3754	confirm
200 Heart Clinics Northwest	47.6491	-117.413	confirm
311 North St	41.02625	-73.7473	unconfirmed
3200 South University Drive	26.08221	-80.2488	unconfirmed
333 Medical Center	34.09118	-118.125	unconfirmed
400 Plumas Care Center	39.12999	-121.615	unconfirmed
4090 Pioneer Parkway, WVC, Utah	40.76669	-111.872	confirm
440 Plumas Care Center	39.13148	-121.615	unconfirmed
45th Medical Group Satellite Pharmacy	28.21853	-80.6025	unconfirmed
480 Plumas Care Center	39.13291	-121.615	unconfirmed
501 Hardy	33.94943	-118.349	confirm
586th Field Hospital	36.64985	-87.4692	unconfirmed
626 409 3568	34.07434	-118.014	unconfirmed
670 Sierra Rose Care Center	39.47305	-119.799	unconfirmed
7th Avenue Pediatrics	26.12454	-80.1501	unconfirmed
801 Wayne Ave.	38.99708	-77.0223	unconfirmed
86th Combat Support Hospital	36.6505	-87.4698	unconfirmed
93 Church St Frederica, DE	39.05932	-75.4047	unconfirmed
969 Plumas Care Center	39.14258	-121.617	unconfirmed
990 Sonoma Avenue Care Center	38.43888	-122.701	unconfirmed
A ChildrenÂ´s Miracle Network Hospital	33.90283	-98.5006	unconfirmed
A Witty Invention	32.32606	-86.2081	unconfirmed
A Woman_s Center-Reproductive	30.38574	-91.0384	confirm
A. Alfred Taubman Health Care Center	42.28394	-83.7278	confirm
A. ELI GABAYAN, M.D. - Beverly Hills Cancer Center	34.06676	-118.386	unconfirmed
A. Mike Tummers, MD	35.9681	-83.1879	unconfirmed
A.V. Bhutwala, M.D.	34.47301	-117.287	unconfirmed
AAMG Annapolis Primary Care	38.99058	-76.5374	unconfirmed
Aarav hospital	40.90426	-74.1618	unconfirmed
AA	31.08765	-97.3449	confirm

Figure 14: Database with detection column to determine which locations have a helipad

Figure 14 displays a snapshot of the database constructed withe name latitude, longitude, and detection columns. This allows a user to input a list of POIs and determine which locations have helipads. Following this naming convention, the Imagery data is also saved with its name as the location lat and lon, this provides an image gallery of helipad locations with their associated name and location.

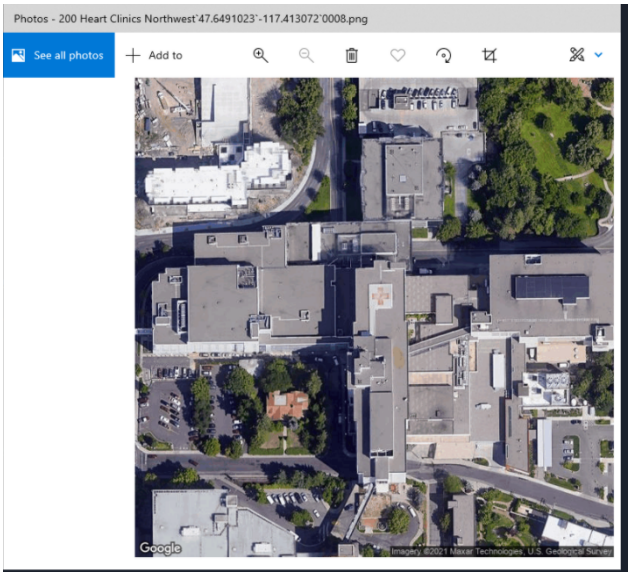

Figure 15: example of imagery data base with name, lat, lon, as the image title

with the identification of helipads a list of these locations could be compiled into one csv, construction a database of known POI's with helipads. This allows for the quick search of over 2133 locations with a helipad detected by the model. It is important to note however there is potential for false positives and negatives as the models accuracy is not perfect. with ongoing model improvement these can be reran through the model to mitigate this problem, as well as manual verification. These steps are currently in progress, and this pipeline could be expanded to other locations of interest if they were desired such has police stations, and other government buildings.

2115	Wolfson Children_s Hospital_ Irazuzta Jose E MD	30.31552	-81.6635	confirm
2116	Wolfson Children_s Hospital	30.31544	-81.6634	confirm
2117	Women and Infants - St. Agnes Hospital	43.77764	-88.4313	confirm
2118	Women_s and Children_s Hospital-Pediatrics Specialty Clinic	30.15244	-92.0464	confirm
2119	Women_s and Children_s Hospital	38.95755	-92.2883	confirm
2120	Women_s Health - Winona Health	44.03295	-91.6233	confirm
2121	Wound Care Center At Clara Maass	40.7856	-74.176	confirm
2122	WVU Medicine Physician Office Center	39.65274	-79.9584	confirm
2123	WVU Medicine	39.65312	-79.9568	confirm
2124	X-Ray Medical Group Radiation	33.12509	-117.075	confirm
2125	Yale New Haven Children_s Hospital	41.30438	-72.9365	confirm
2126	Yale New Haven Hospital Saint Raphael Campus	41.31074	-72.9433	confirm
2127	YMRC Diagnostic Imaging	32.6824	-114.635	confirm
2128	YNHH Shoreline Guilford	41.29115	-72.6639	confirm
2129	Yoga for Pregnancy at Pomerado Hospital	32.99641	-117.054	confirm
2130	Your Fat Loss Coach	26.12188	-80.1416	confirm
2131	Yuma Regional Medical Center	32.68232	-114.635	confirm
2132	Zenith Family Health	40.43031	-111.85	confirm
2133	ZoomCare Super - Lloyd District	45.52742	-122.661	confirm

Figure 16: Snap shot of the database with all the confirmed locations

Figure 16 displays the csv file with all the confirmed locations in a single table. this is essentially the POI database, a collection of all the POI helipads detected so far. This is useful for quickly

determining if a given location has a helipad with a simple search, filtering out all the unconfirmed locations to reduce the size.

7 Conclusion

We developed a deep learning model for helipad identification and detection from aerial Google Earth imagery. We also devised a framework to begin searching for helipads in designated areas. This framework was developed into a pipeline of Data acquisition, model training, and then leveraging the model to acquire more data. The Helicheck web app was developed to provide an accessible and easy to use platform for fast identification of a helipad in a given location. Then a database of all the helipads found so far was created, as well as a method for inputting a table of locations with lat and lon and determining which of these locations have a helipad in them. Further work is being conducted to improve upon the sliding window approach, and the continuation of this pipeline can be expanded to various other POIs to further expand this dataset, ultimately creating a comprehensive data base of helipads in the US and potentially beyond.

References

[FAA, 2012] FAA (2012). 150/5390-2c: Heliport design - faa standards for the design of heliports serving helicopters with single rotors.

[LeCun et al., 2015] LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *Nature*, 521:436–444.

[Patruno et al., 2017] Patruno, C., Nitti, M., Stella, E., and D'Orazio, T. (2017). Helipad detection for accurate UAV pose estimation by means of a visual sensor. *International Journal of Advanced Robotic Systems*, 14(5).

[Pierre et al., 2018] Pierre, Z., Mavromatis, S., Sequeira, J., Anoufa, G., Belanger, N., and Fillias, F.-X. (2018). Embedding intelligent image processing algorithms: the new safety enhancer for helicopters missions. In *44th European Rotorcraft Forum-ERF 2018*.

[Prakash and Saravanan, 2016] Prakash, R. O. and Saravanan, C. (2016). Autonomous robust helipad detection algorithm using computer vision. In *International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT)*, pages 2599–2604.

[Rungta et al., 2020] Rungta, A., Soni, Y., Agarwal, P., Ghosh, B., and Kumar, S. (2020). Real-time and autonomous detection of helipad for landing quad-rotors by visual servoing.

[Selvaraju et al., 2019] Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., and Batra, D. (2019). Grad-cam: Visual explanations from deep networks via gradient-based localization. *International Journal of Computer Vision*, 128(2):336–359.

[Simonyan et al., 2014] Simonyan, K., Vedaldi, A., and Zisserman, A. (2014). Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv*, 1312.6034.

[Walker, 2019] Walker, G. (2019). No one knows where America's helipads are, except this neural network.

[Zdenek Kalal, Krystian Mikolajczyk, and Jiri Matas, 2010] Zdenek Kalal, Krystian Mikolajczyk, and Jiri Matas (2010). Forward-Backward Error: Automatic Detection of Tracking Failures. In *International Conference on Pattern Recognition*, page 2756–2759.

A Sample Results

A.1 True positives

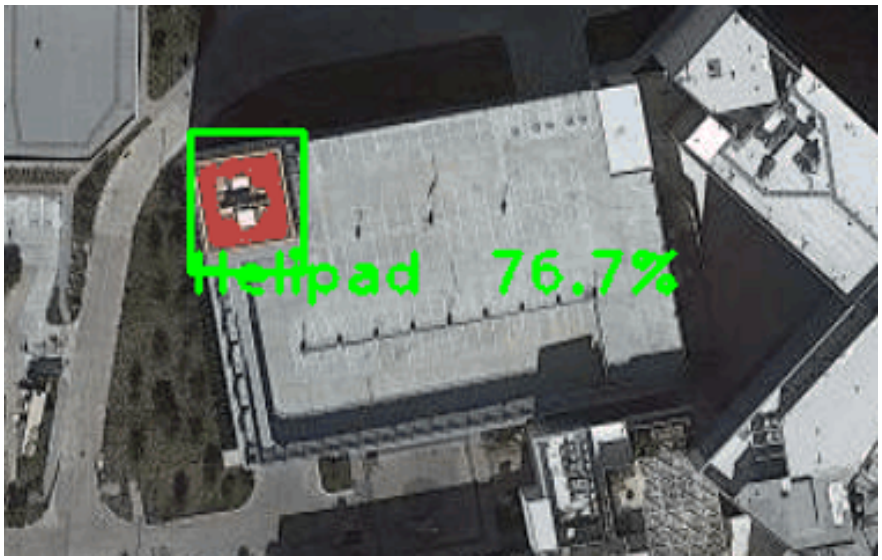

Figure 17: example of a true positive

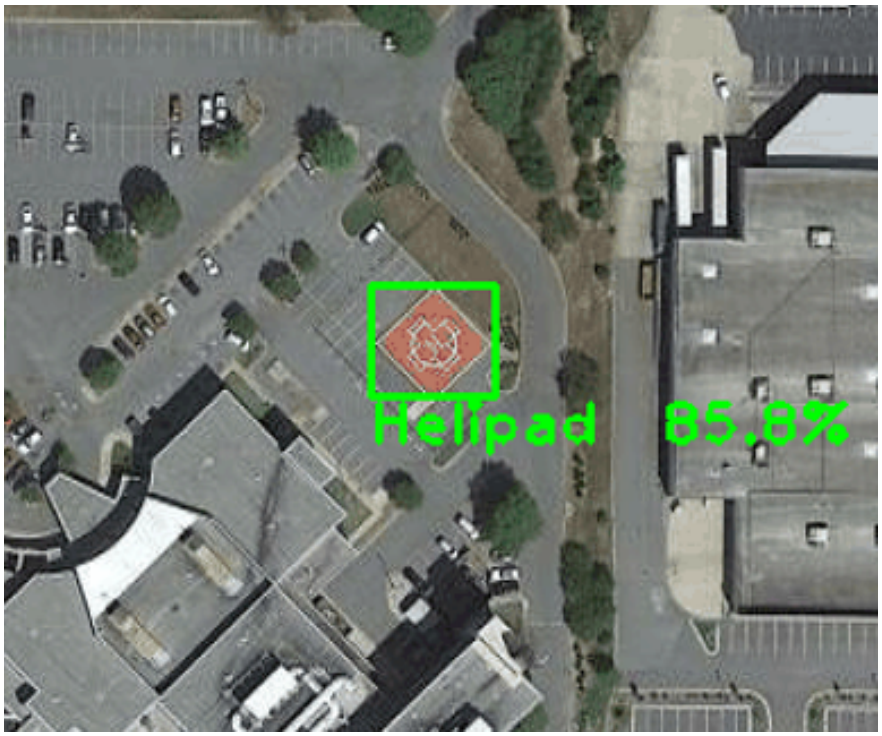

Figure 18: example of a true positive

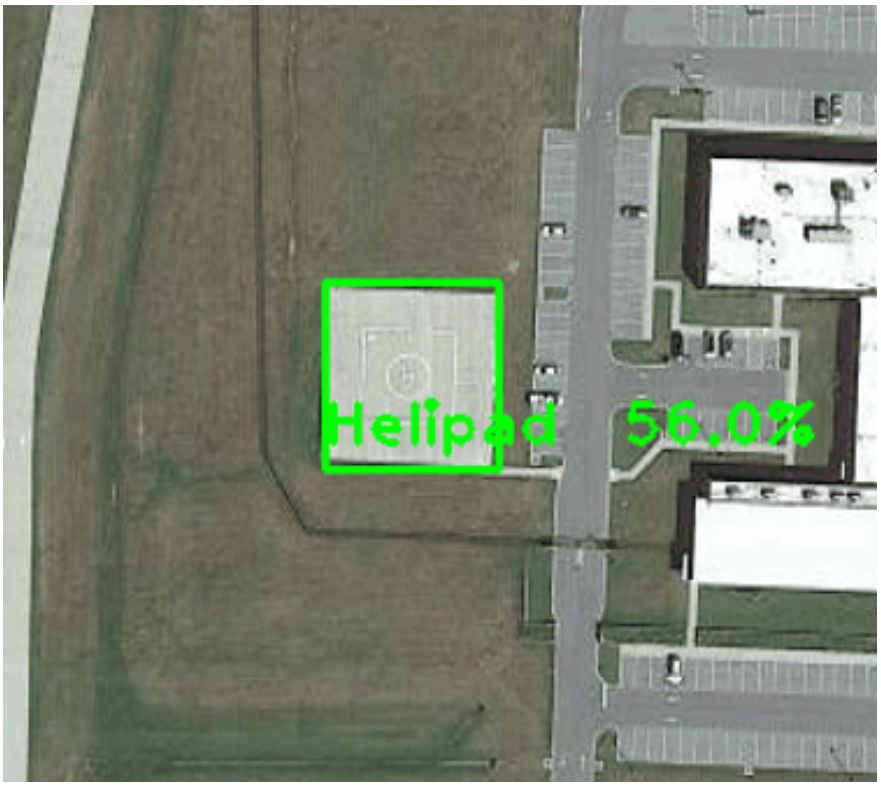

Figure 19: example of a true positive

Figure 20: example of a true positive

Figure 21: example of a true positive

A.2 False Negatives

Figure 22: example of a false positive

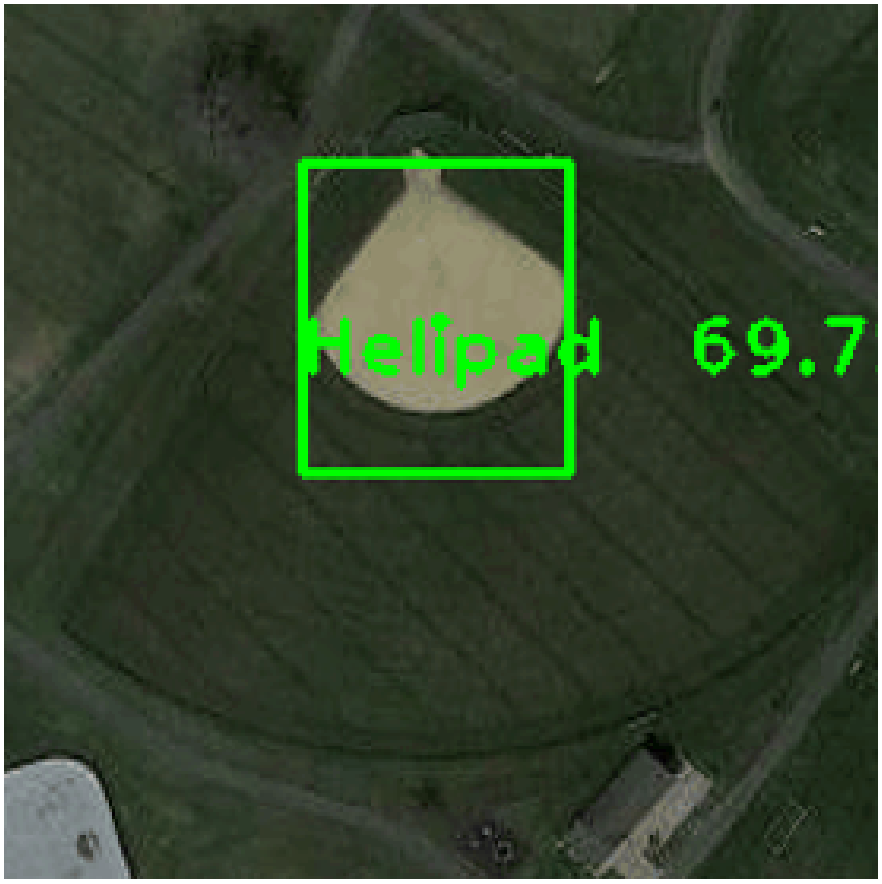

Figure 23: example of a false positive

Figure 24: example of a false positive

Figure 25: example of a false positive

Figure 26: example of a false positive

Figure 27: example of a false positive

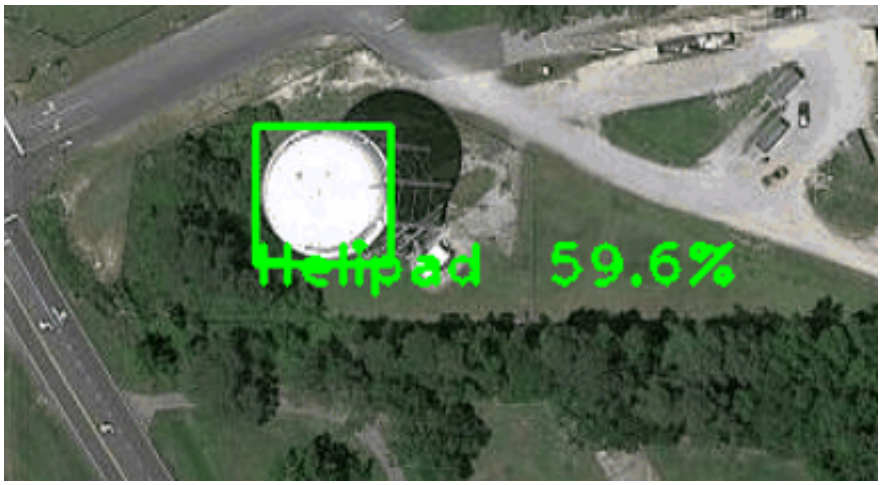

Figure 28: example of a false positive

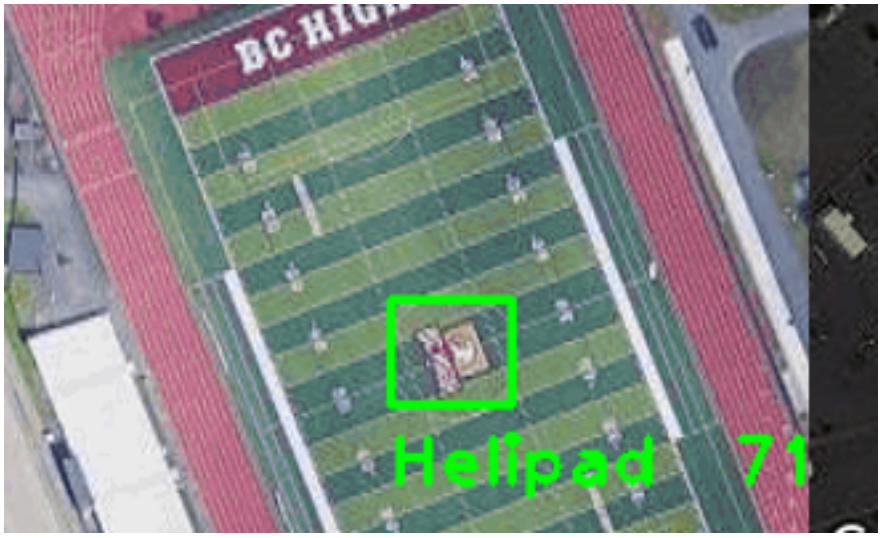

Figure 29: example of a false positive