Rotorcraft Landing Sites – An AI-Based Identification System

FINAL REPORT
05-2021

Submitted by:

Ghulam Rasool
Assistant Professor

Nidhal Bouaynaya
Professor

Mohammad Jalayer
Assistant Professor

David Specht
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-REG 32		

4. Title and Subtitle	5. Report Date
Rotorcraft Landing Sites – An AI-Based Identification System	02 / 2021
	6. Performing Organization Code CAIT/Rowan

7. Author(s)	8. Performing Organization Report No.
Ghulam Rasool https://orcid.org/0000-0001-8551-0090, Nidhal Bouaynaya https://orcid.org/0000-0002-8833-8414, Mohammad Jalayer https://orcid.org/0000-0001-6059-3942, David Specht https://orcid.org/0000-0002-1894-2034	CAIT-UTC-REG 32

9. Performing Organization Name and Address	10. Work Unit No.
Henry M. Rowan College of Engineering, Rowan University, 201 Mullica Hill Road, Glassboro, NJ 08028	11. Contract or Grant No. 69A3551847102

12. Sponsoring Agency Name and Address	13. Type of Report and Period Covered
Center for Advanced Infrastructure and Transportation Rutgers, The State University of New Jersey 100 Brett Road Piscataway, NJ 08854	Final Report 01/1/2020 – 12/31/2020
	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
Location data about U.S. heliports is often inaccurate or nonexistent in the FAA's databases, which leaves pilots and air ambulance operators with erroneous information about where to find safe landing zones. In the 2018 FAA Reauthorization Act, Congress required the FAA to collect better information from the helicopter industry under part 157, which covers the construction, alteration, activation, and deactivation of airports and heliports. At the same time, there is no requirement to report private helipads to the FAA when constructed or removed, and some public heliports do not have up-to-date records. This project developed an autonomous system that can authenticate the coordinates present in the FAA master landing site database. Our system can search for helipads in designated large areas around the country. The proposed approach is based on a convolutional neural network model that learns optimal helipad features from the data. We used the FAA's 5010 database and others to construct a benchmark database of rotorcraft landing sites. The database consists of 9,324 aerial images containing helipads, helistops, helidecks, and helicopter runways in rural and urban areas. The database also includes negative examples, i.e., satellite images, e.g., rooftop of buildings and fields that are not designated landing sites. The dataset was used to train various state-of-the-art convolutional neural network models (CNN). The outperforming model, EfficientNet-b0, achieved nearly 95% validation accuracy.

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	39	

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.

2 Convolutional Neural Networks (CNNs)

2.1 Learning Features with 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

Figure 1: 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.

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.

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 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 1 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 1. The difference between the two zoom levels can be seen in Figure 2. 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.

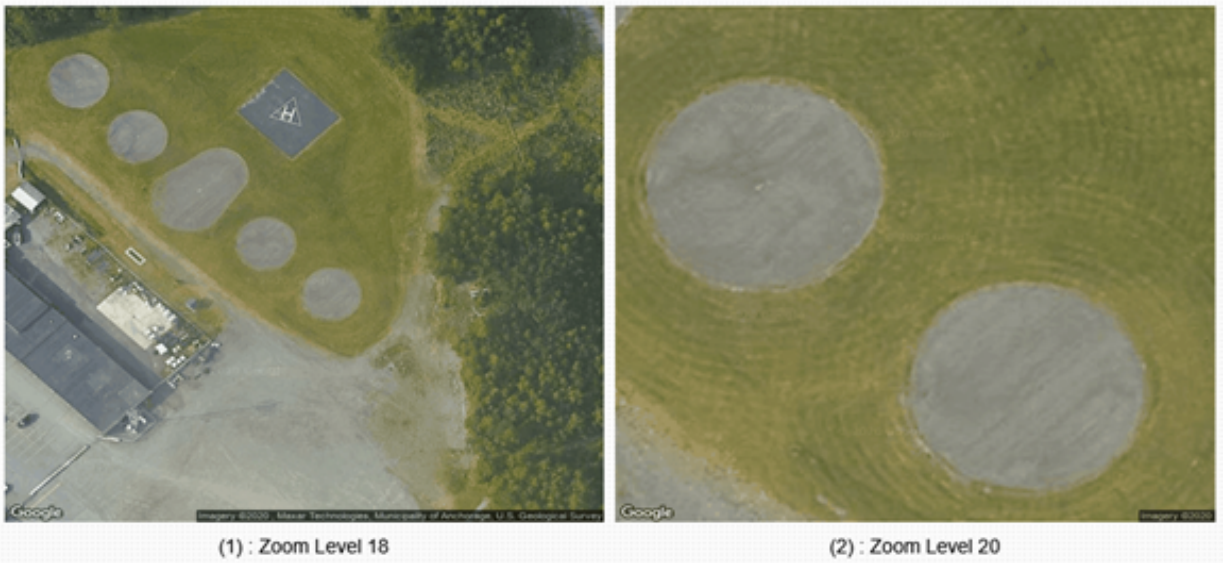

Figure 2: Zoom level in Google Map API. Left image is downloaded at the zoom level of 18 and the right image at the zoom level of 20. At a zoom of 18, numerous possible parking pads and a helipad are visible. The helipad is not visible at the zoom of 20.

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 1, 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 Final 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.

The final benchmark dataset has $9,324$ satellite images labeled as either helipad or non-helipad. Figure 3 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 3. 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.

Figure 3: 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.

4 Explainable Identification of Helipads

4.1 Convolutional Neural Network Models

We considered four different CNN models for the helipad detection task. These included ResNet101, Inception, Xception, and EfficienetNet-B0 [Tan and Le, 2019, He et al., 2016] [Szegedy et al., 2016, Chollet, 2017]. These models were chosen as they represent a variety of families of common CNN architectures. The details of the these models are presented in Table 1. All of these models served as feature extractors with additional layers for fine-tuning and classification as shown in Figure 4.

Table 1: Comparison of selected CNN models

Model	Skip	Inception	Parameters (10^6)	Image-net Top 5-accuracy
ResNet101	yes	no	44.71	92.8%
Inception-V3	no	yes	23.85	93.7%
Xception	yes	no	22.91	94.5%
EfficientNet-b0	yes	no	5.33	97.1%

4.1.1 ResNet101

ResNet101, proposed in [He et al., 2016], is a residual network that contains skip connections. Residual networks were designed to mitigate the problem of *accuracy degradation with network depth*. With the network depth increasing, accuracy gets saturated and then degrades rapidly. Skip connections simply perform identity mapping, and their outputs are added to the outputs of the

stacked layers. It was experimentally shown that deep residual networks: 1) exhibit lower training error when the depth increases compared to their counterpart "plain" networks, and 2) enjoy accuracy gains from considerably increased depth, producing significantly better results than their counterpart "plain" networks.

4.1.2 Inception-V3

Inception-V3, proposed in [Szegedy et al., 2016], is part of the family of *Inception* networks. This family of networks uses the Inception module, which leverages multiple types of filter sizes in a convolutional layer instead of being restricted to single filter size. The motivation of Inception stems from the human visual cortex, which identifies patterns at different scales and then accumulates them to form larger perceptions of objects. Therefore, Inception modules have the potential to improve optimal feature extraction, and hence improve learning.

4.1.3 Xception

Xception, proposed in [Chollet, 2017], improves upon the Inception family of architectures by replacing Inception modules with depth-wise separable convolutions. A depth-wise separable convolution is a spatial convolution performed independently over each channel of an input, followed by a point-wise convolution, i.e., a 1×1 convolution, projecting the channels output onto a new channel space. Xception is built by stacking depth-wise separable convolutions. This model also uses skip connections.

4.1.4 EfficientNet-b0

The EfficientNet family of architectures was proposed in [Tan and Le, 2019] to address the issue of scaling CNN models for better accuracy. Based on a compound scaling method that balances network width, depth, and resolution, eight different models (b0 - b7) were proposed, where higher values correspond to larger networks. The models are made up of sections of repeating layers that can be efficiently scaled to create a deeper model.

5 Experimental Results

We performed 10-fold validation for each network. The results from each of the 10 runs were averaged to produce the average performance of the model as shown in Figure 5. According to Figure 5, the EfficientNet-b0 model performed the best on our benchmark helipad dataset.

Figure 6 shows the results from the grad-CAM implementation. This overlay shows a heat-map of the most salient pixels in the model's prediction (EfficientNet-b0). Observe that the network

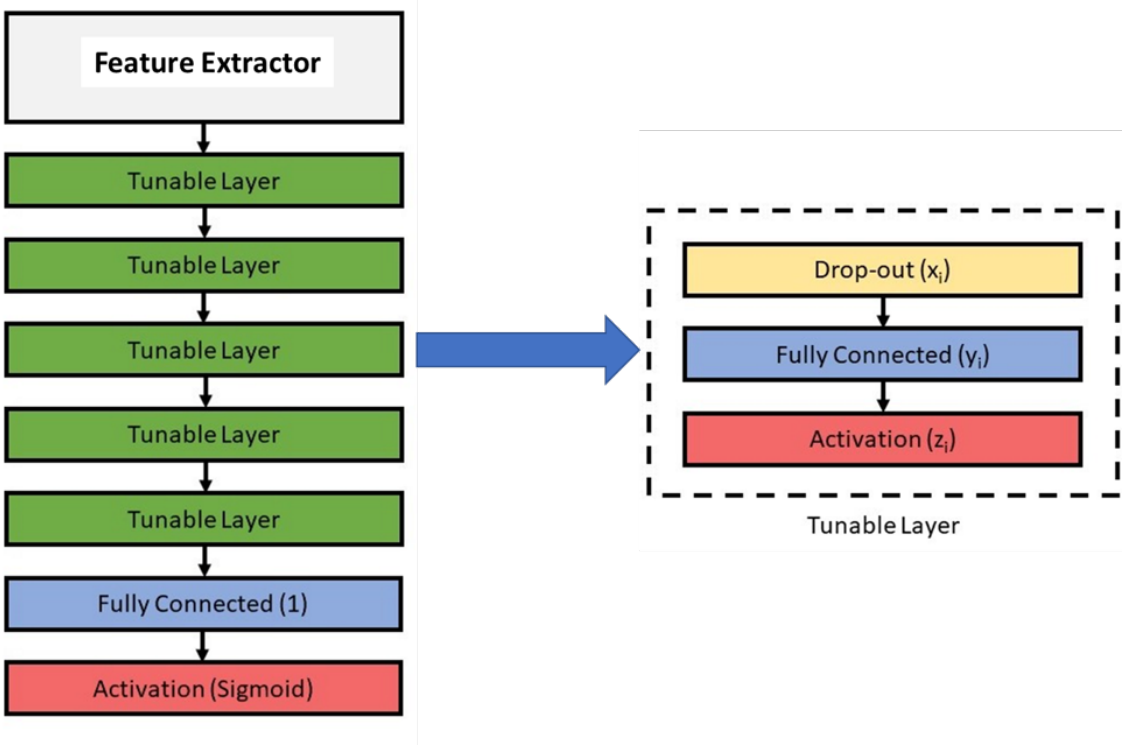

Figure 4: The layout of the proposed machine learning model is presented. The left image shows an overall structure of the CNN starting from the top where the satellite image is input to the CNN. The right side of the figure shows the inner structure of all tunable layers that make up the classifier part of the proposed CNN.

relied on features of the helipad to make its prediction, as opposed to background features, such as nearby buildings.

6 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.

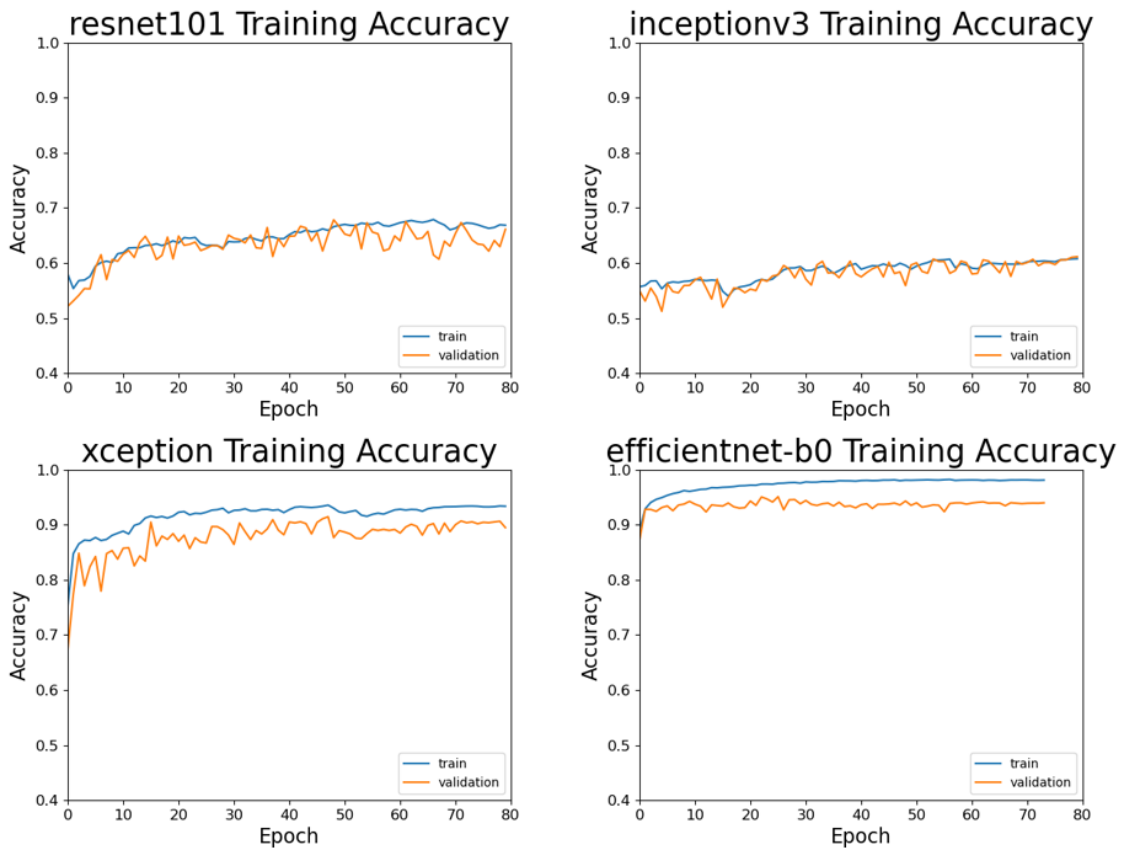

Figure 5: Training (blue) and validation (orange) accuracy curves for ResNet101 (top left), Inception-V3 (top right), Xception (bottom left), and EfficientNet-b0 (bottom right).

6.1 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.

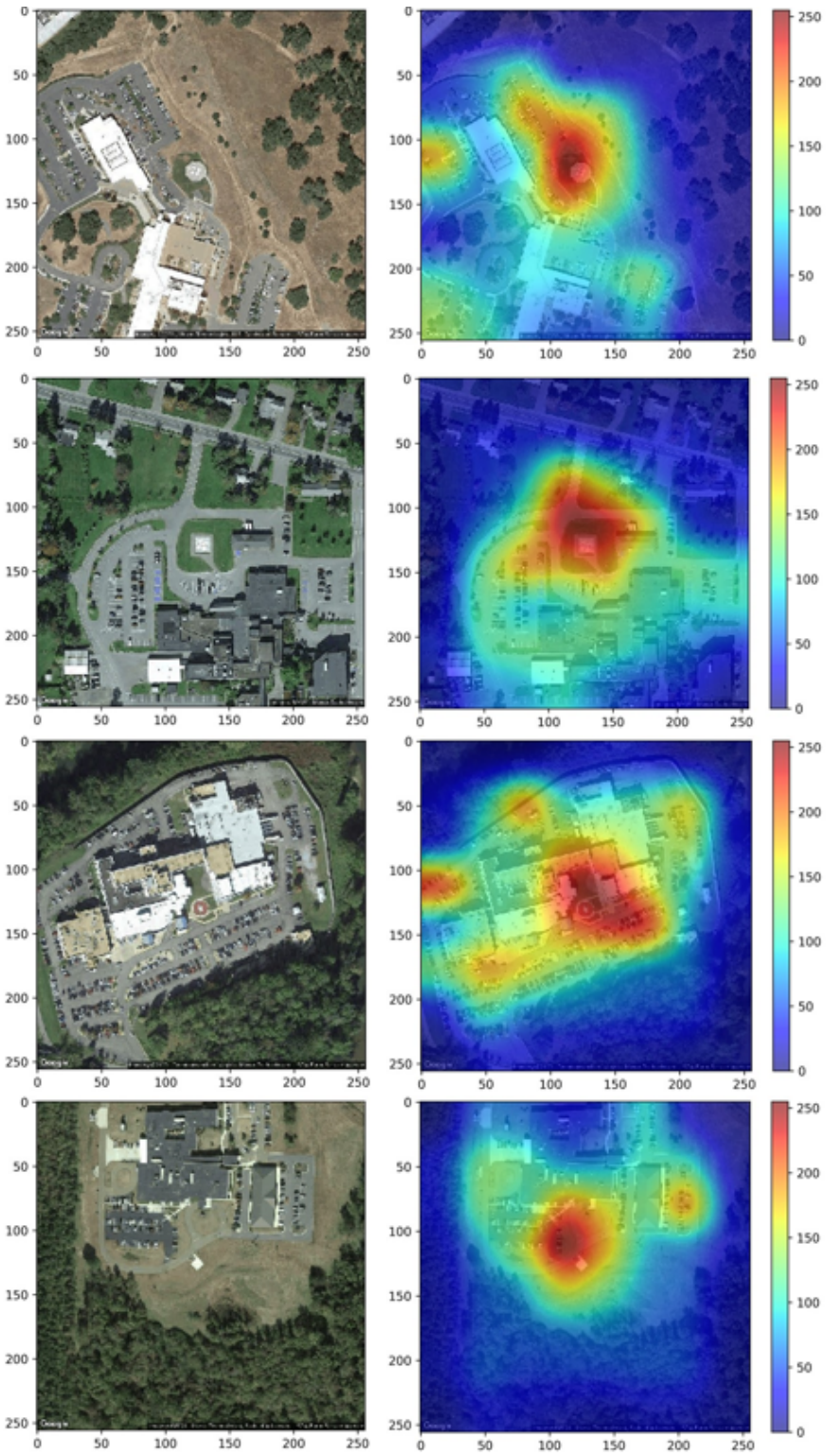

Figure 6: Grad-CAM heat-maps showing the importance of pixels in the CNN model (EfficientNet-b0) prediction. Aerial images (left column) and their corresponding Grad-CAM heat-maps (right column). The red area refers to the part of the model where the network attention is strong, and the blue part refers to the part that does not influence the prediction.

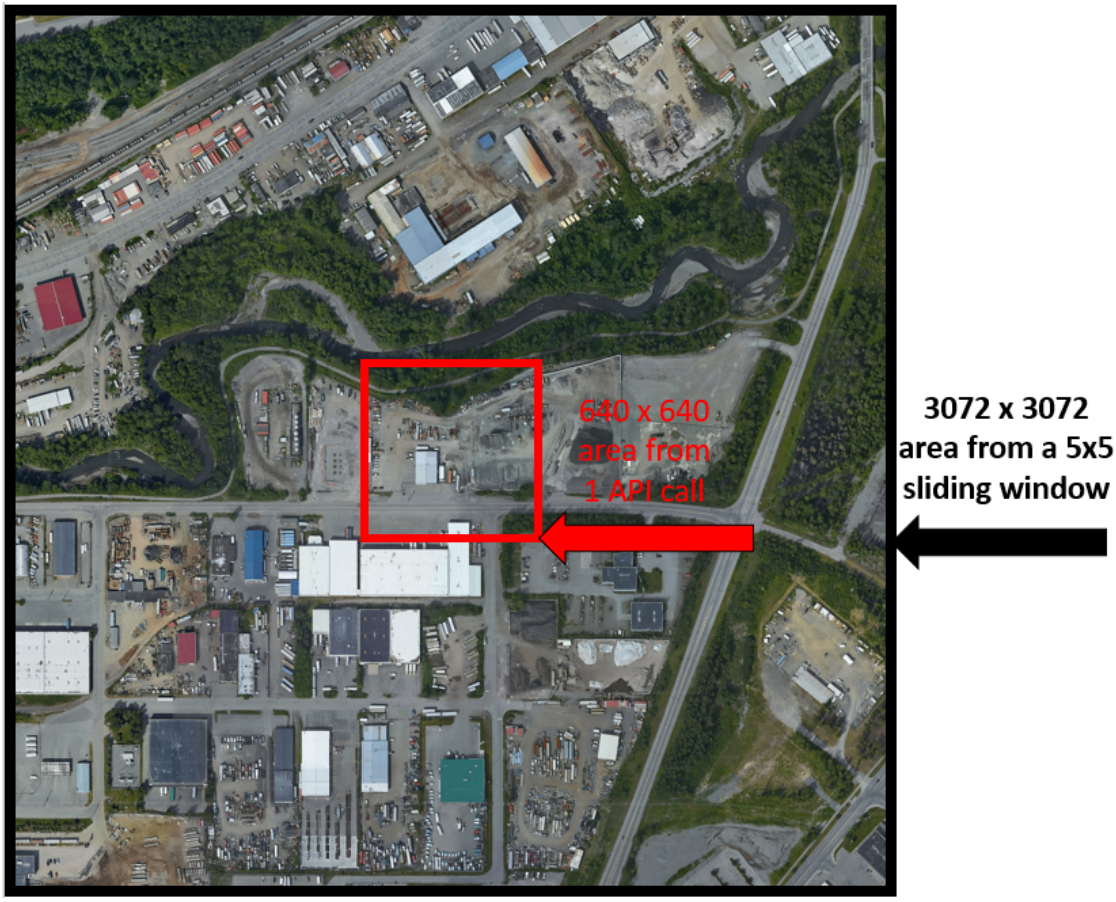

Figure 7: Searching helipads in large ares. The area scanned in a 5×5 collage settings vs. a single API call.

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 7. 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.

6.2 Searching for Helipads in Los Angeles

We applied this collage technique to create a Los Angeles (LA) region. LA makes for an interesting testing area, as it has a high helipad density, so there will be many helipads to detect in a small region. However, the LA region is notably different than the other cityscapes in the dataset. To fix this data imbalance, some of the data from LA was used to supplement the benchmark dataset. Figure 8 shows the LA area under study. It was formed via an 11×11 collage, and will be broken up using a 20×20 sliding window producing 400 smaller images. The 200 images making up the top half of the image will be part of the supplemental training set, and the 200 images in the bottom half of the image will make up the testing case. The test achieved the following performance measures: Accuracy $= 76.0\%$, Precision $= 61.6\%$, and Recall $= 97.4\%$.

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. We achieved good performance in detecting helipads in the LA region; Nonetheless, more experimentation is needed before this approach is ready for more widespread testing. Notably a larger variety of data should be considered, and a wide variety of locations should be incorporated for testing to ensure that the algorithm will perform in these different locations.

Although the study was limited to the US, this approach is readily extendable to helipad identification and detection across the globe. Future work includes leveraging Grad-CAM interpretability maps to estimate the location of the helipads after their identification.

Figure 8: 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.

Figure 9: Large Area Helipad Search with the Sliding Window Approach. (Left) The test region consists of the lower half of the LA area in Figure 8. (Center) Ground Truth helipad locations. (Right) Model prediction of helipad locations.

References

[Chollet, 2017] Chollet, F. (2017). Xception: Deep learning with depthwise separable convolutions. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1800–1807.

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

[He et al., 2016] He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

[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.

[Szegedy et al., 2016] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826.

[Tan and Le, 2019] Tan, M. and Le, Q. (2019). Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning*, pages 6105–6114. PMLR.

[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 Positive Cases

True Positives (TP) are samples where there is a helipad present in the image, and the AI system predicts that there is a helipad in the image. This is the case where the CNN is capable of correctly identifying a helipad. For an AI system to be considered 100% accurate, then all images containing helipads should be this case, and all images without helipads should be in the True Negative (TN) case. The accuracy of the CNN can be expressed using these terms and total samples in the equation shown below.

$$Accuracy = \frac{TP + TN}{Total\ Samples} \tag{4}$$

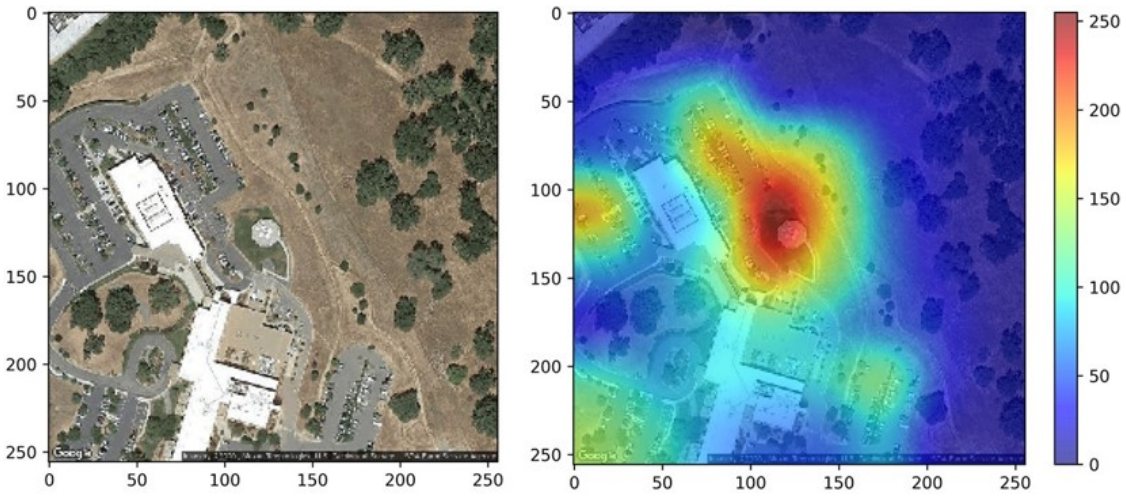

Figure 10: A true positive example from the test dataset, i.e., the AI system identified the helipad correctly. (Left panel): Original image. (Right panel): Explainability map showing parts of the input image used by the AI system to decide about the helipad's presence. Red color shows stronger contribution and blue color weaker.

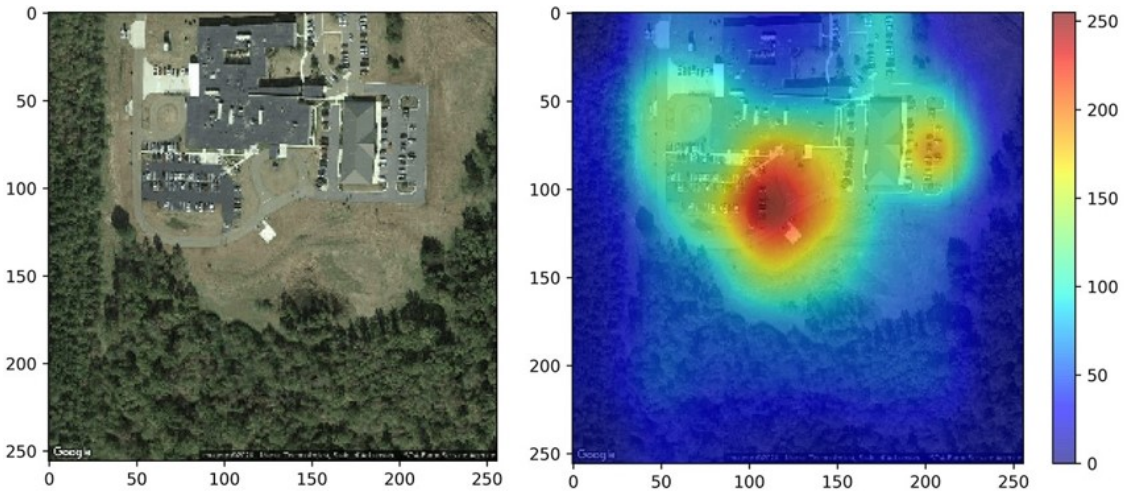

Figure 11: A true positive example from the test dataset.

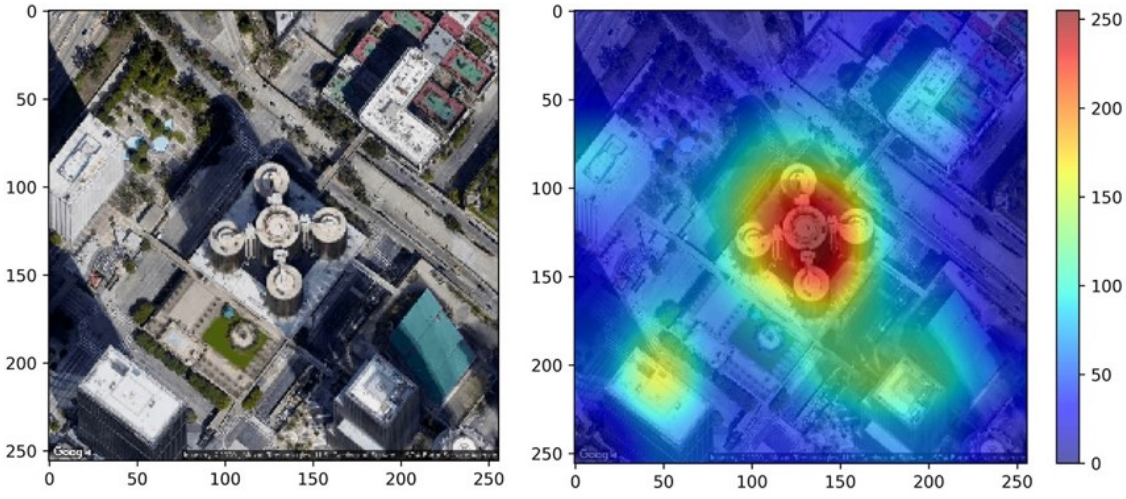

Figure 12: A true positive example from the test dataset.

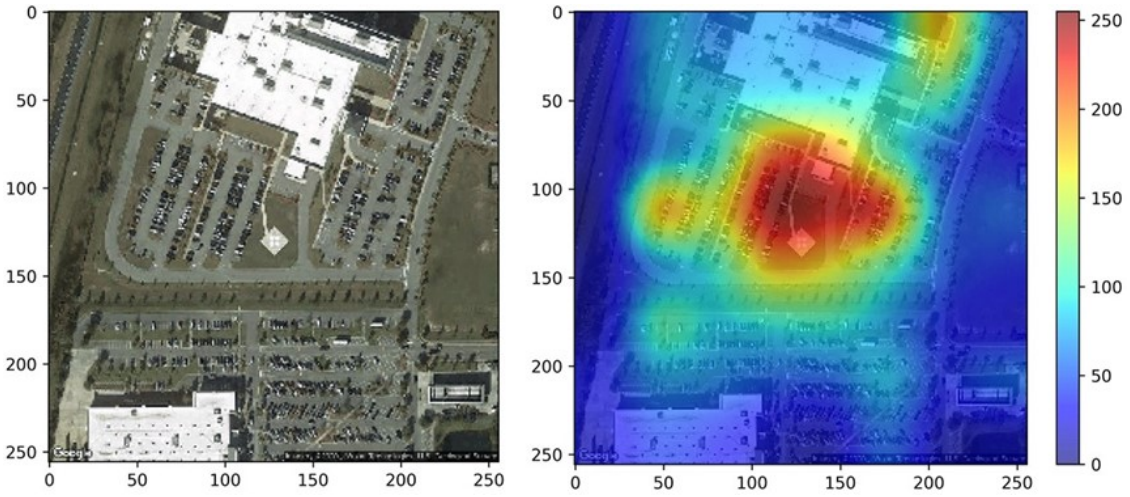

Figure 13: A true positive example from the test dataset.

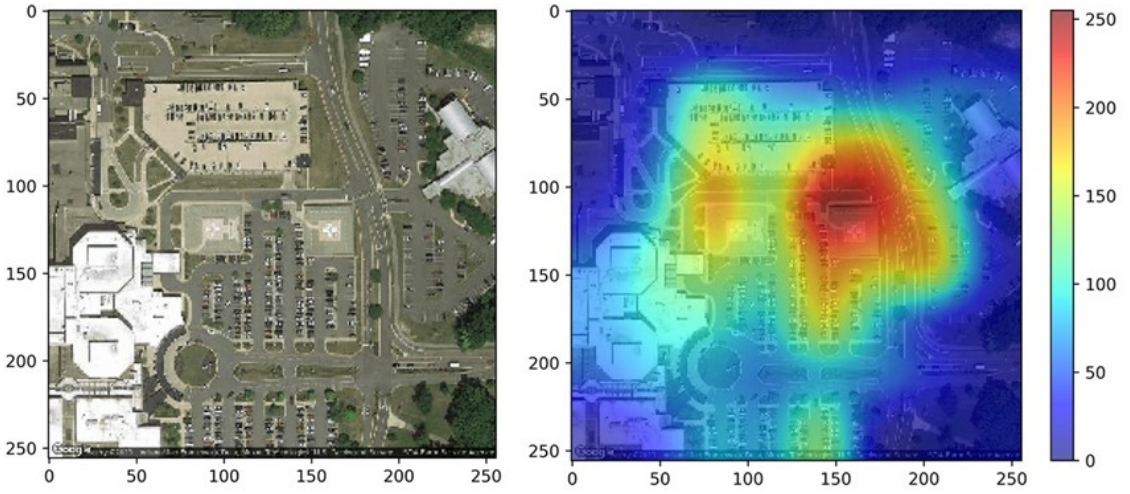

Figure 14: A true positive example from the test dataset.

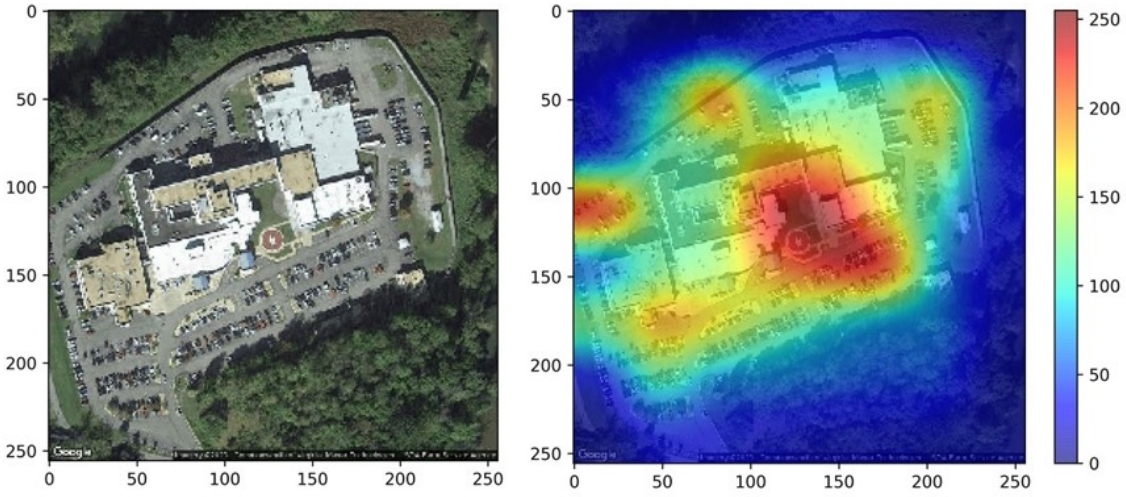

Figure 15: A true positive example from the test dataset.

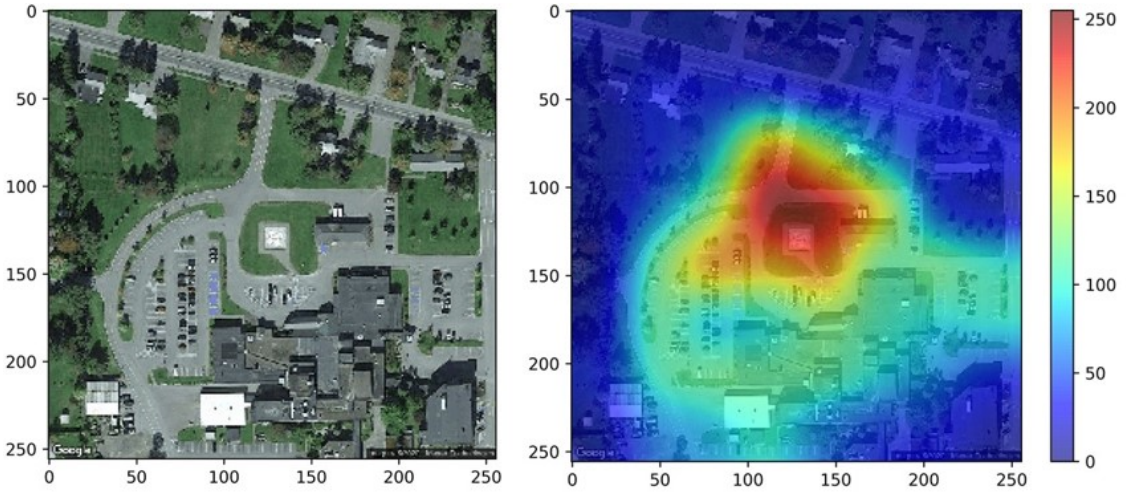

Figure 16: A true positive example from the test dataset.

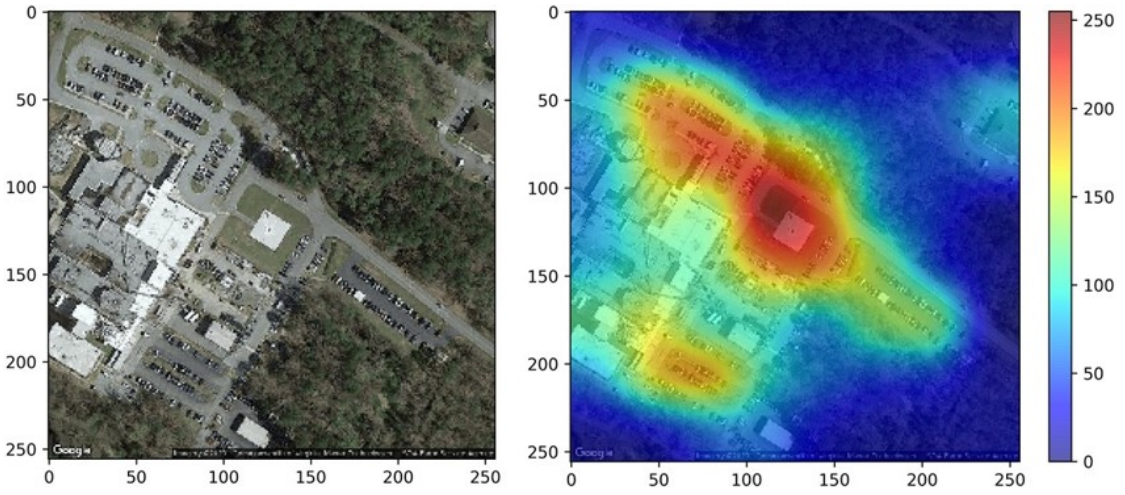

Figure 17: A true positive example from the test dataset.

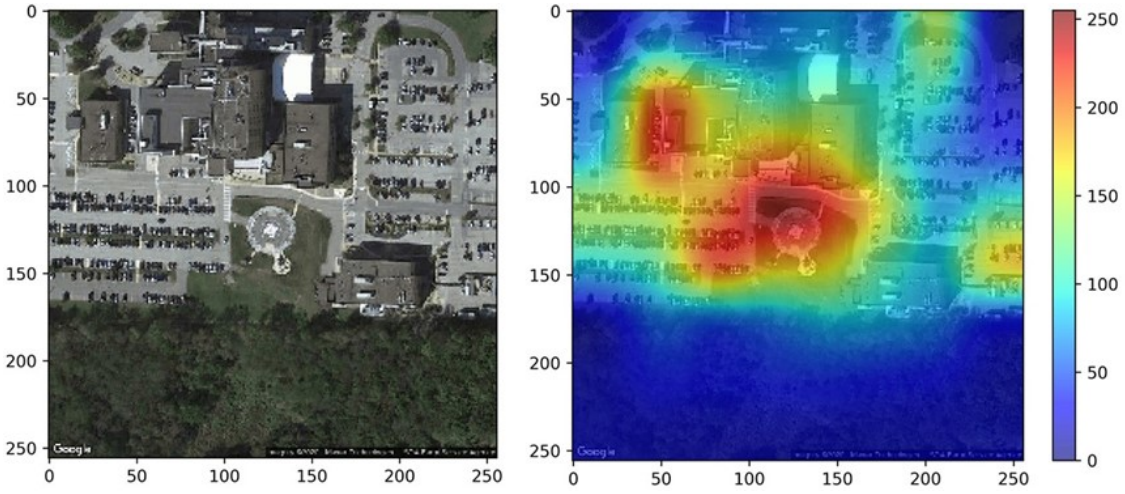

Figure 18: A true positive example from the test dataset.

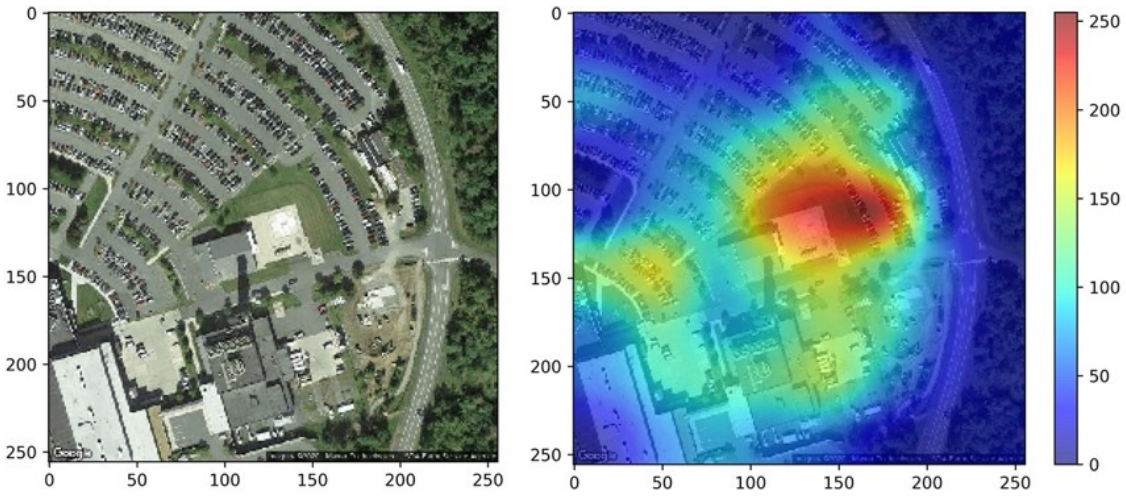

Figure 19: A true positive example from the test dataset.

A.2 False Positive Cases

False positive (FP), also known as type 1 error, is the case where the sample contains no helipad, however the AI system predicts there to be a helipad. The FPs are the cases where the CNN failed, and insights for the reasons the CNN failed can be found from inspecting these cases. FP along with the TP can be used to determine the precision of the system using the equation below. Precision represents the accuracy of the system's predictions when a helipad is present in the given image.

$$Precision = \frac{TP}{TP + FP} \tag{5}$$

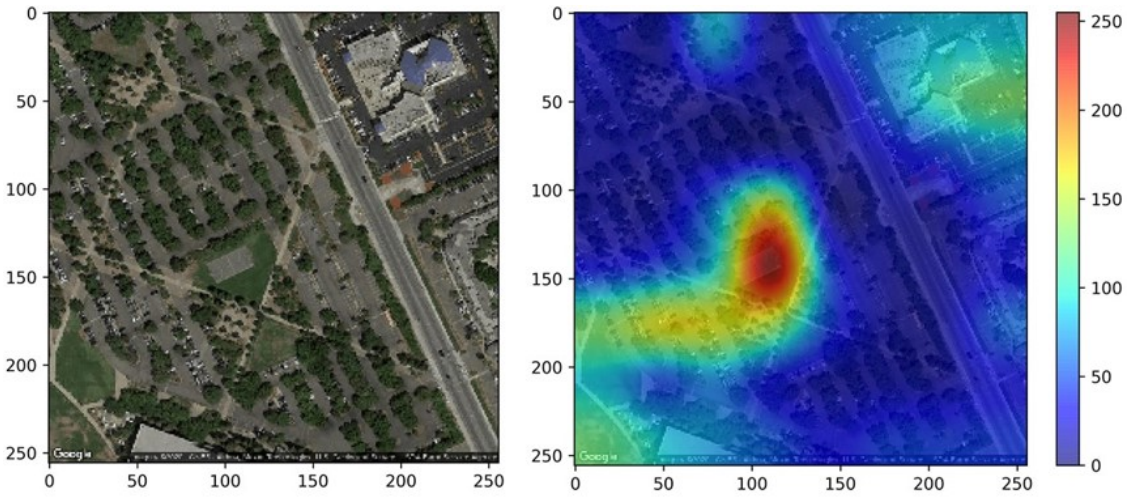

Figure 20: A false positive example from the test dataset, i.e., the AI system identified the helipad, however there was not helipad in the original input image. (Left panel): Original image. (Right panel): Explainability map showing parts of the input image used by the AI system to decide about the helipad's presence. Red color shows stronger contribution and blue color weaker.

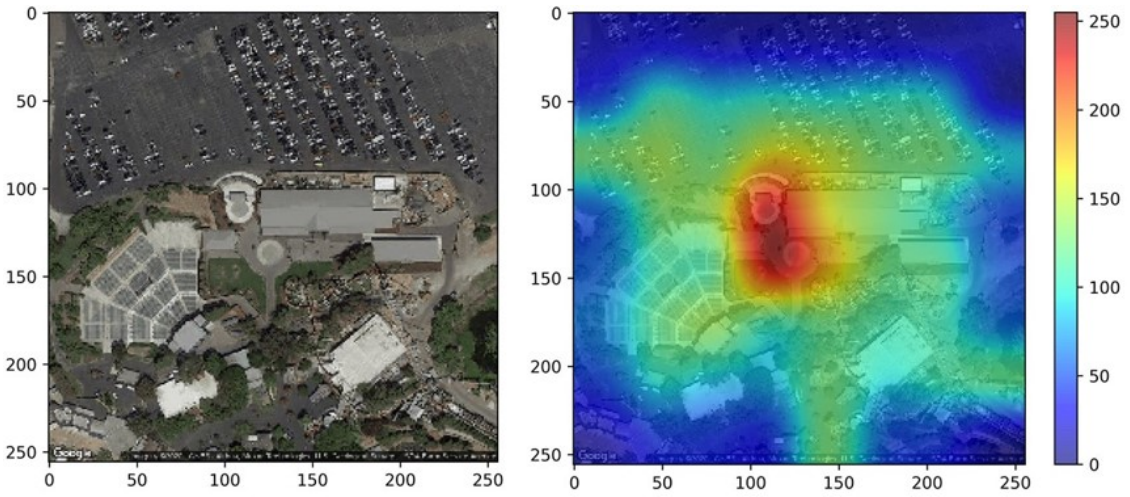

Figure 21: A false positive example from the test dataset.

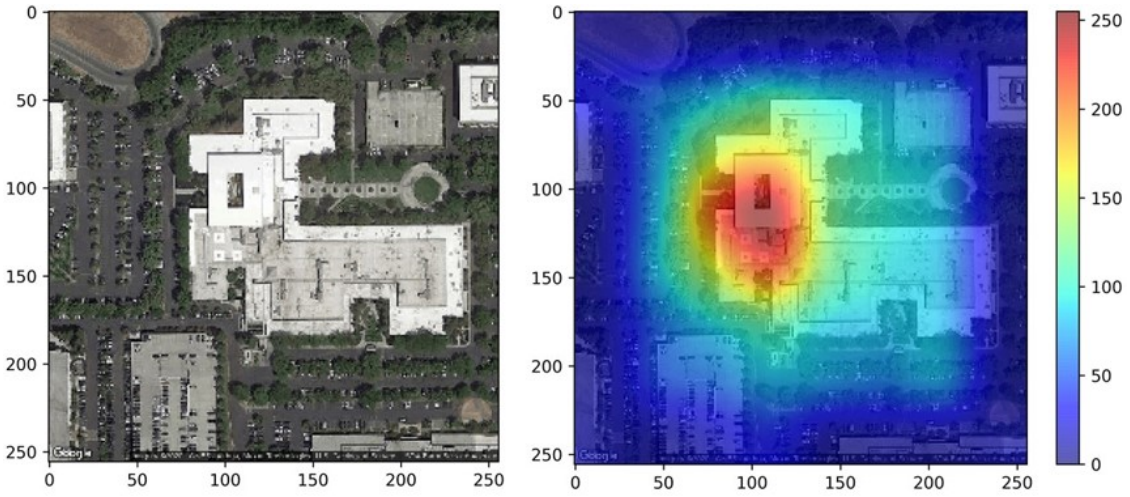

Figure 22: A false positive example from the test dataset.

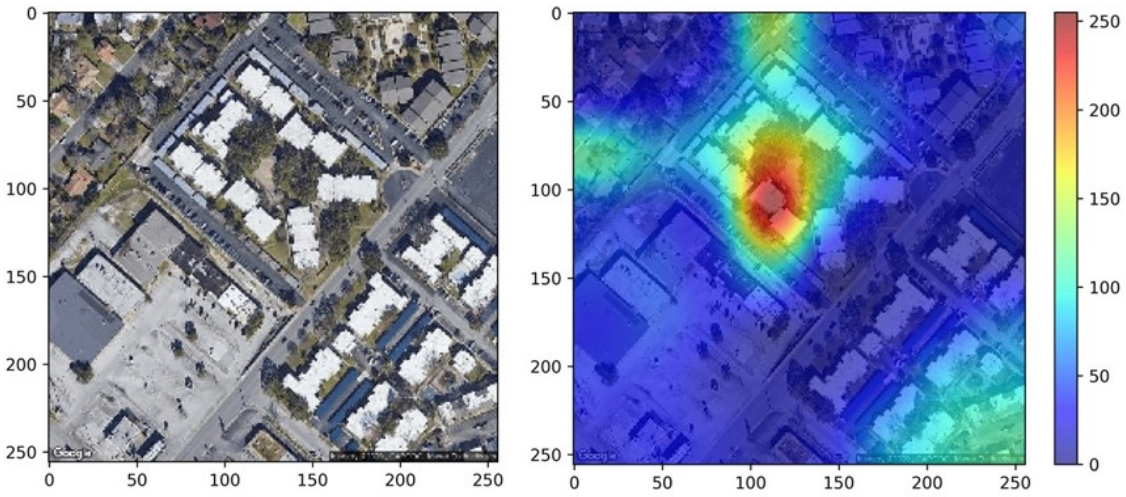

Figure 23: A false positive example from the test dataset.

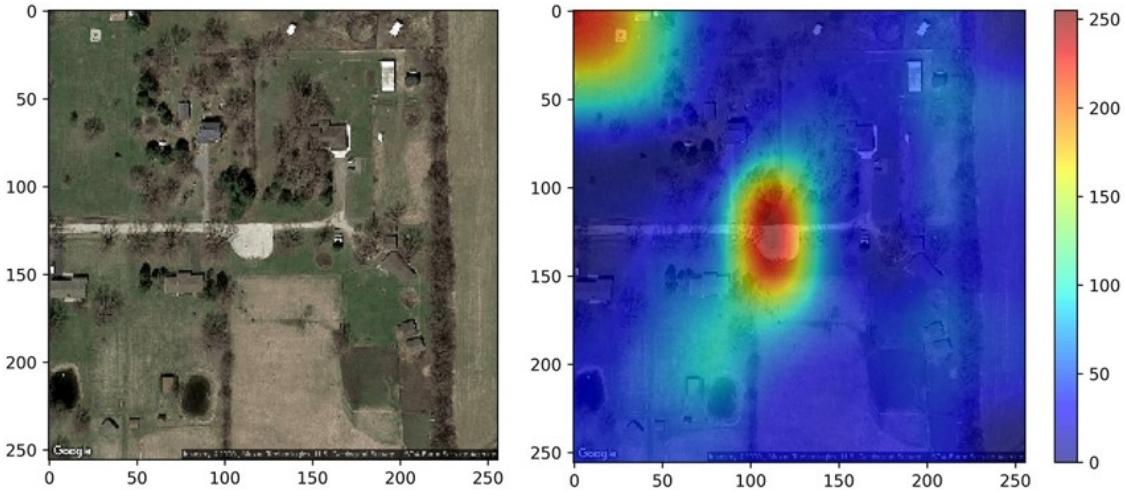

Figure 24: A false positive example from the test dataset.

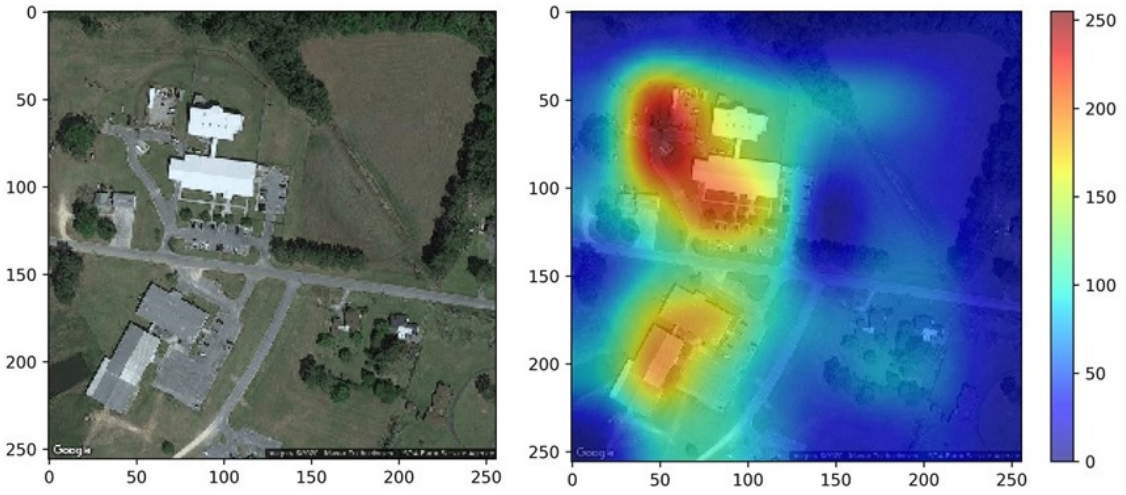

Figure 25: A false positive example from the test dataset.

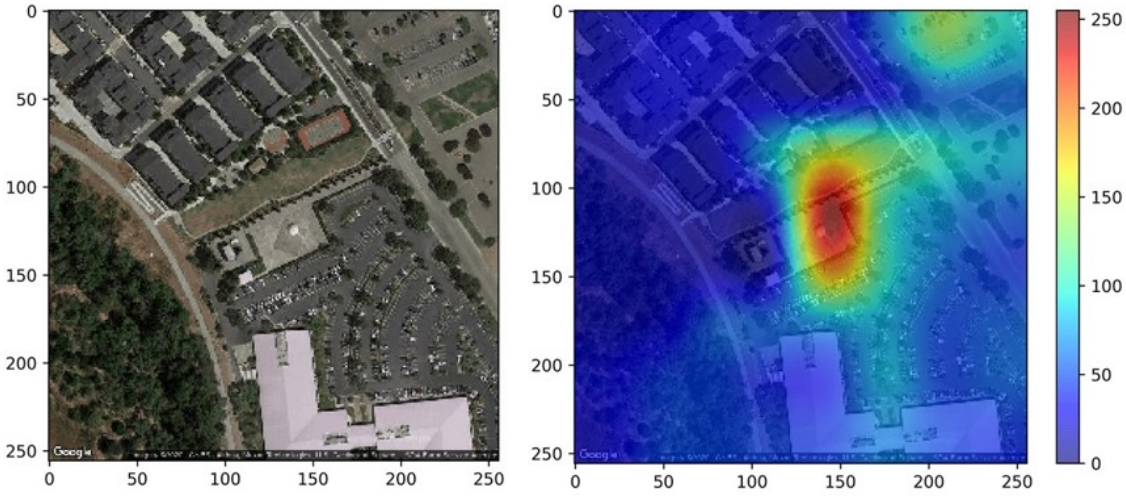

Figure 26: A false positive example from the test dataset.

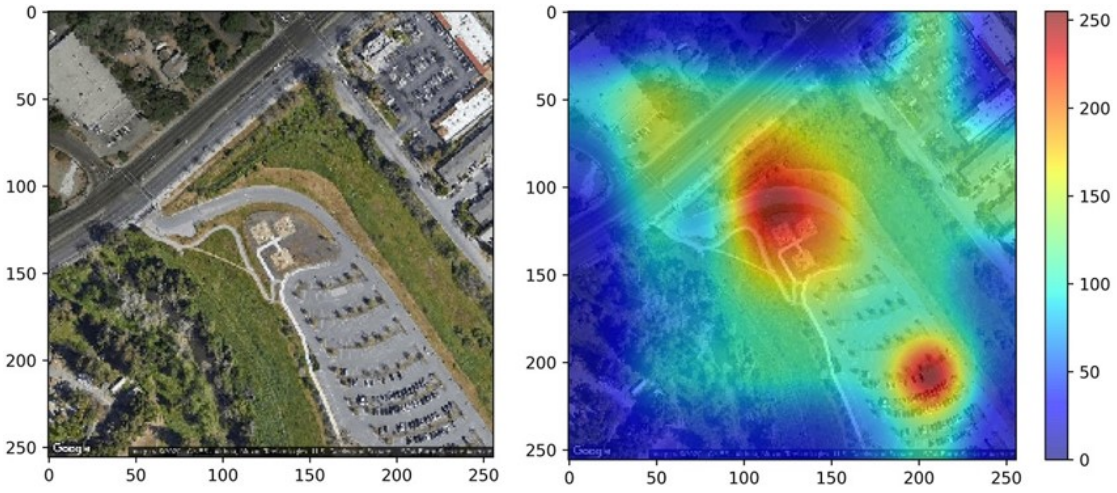

Figure 27: A false positive example from the test dataset.

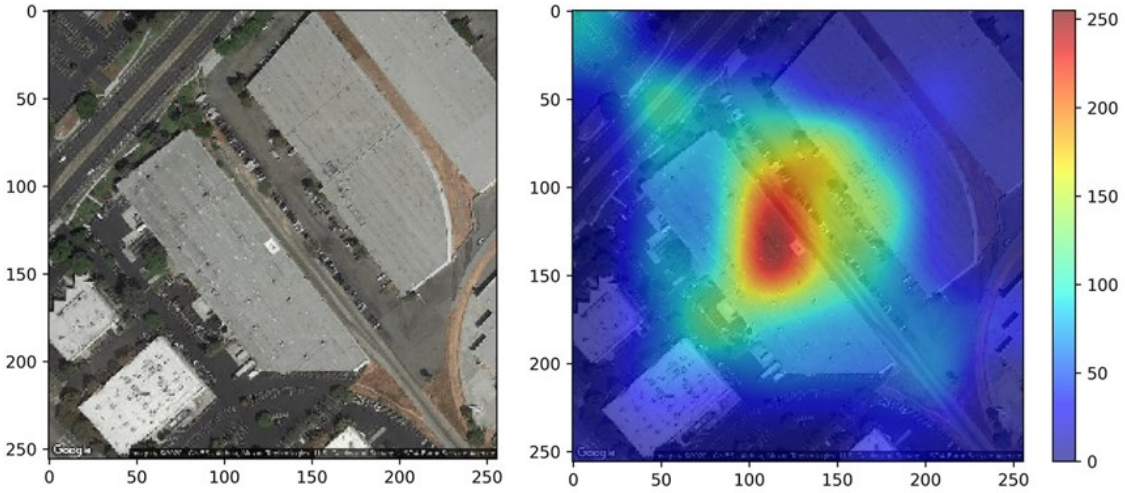

Figure 28: A false positive example from the test dataset.

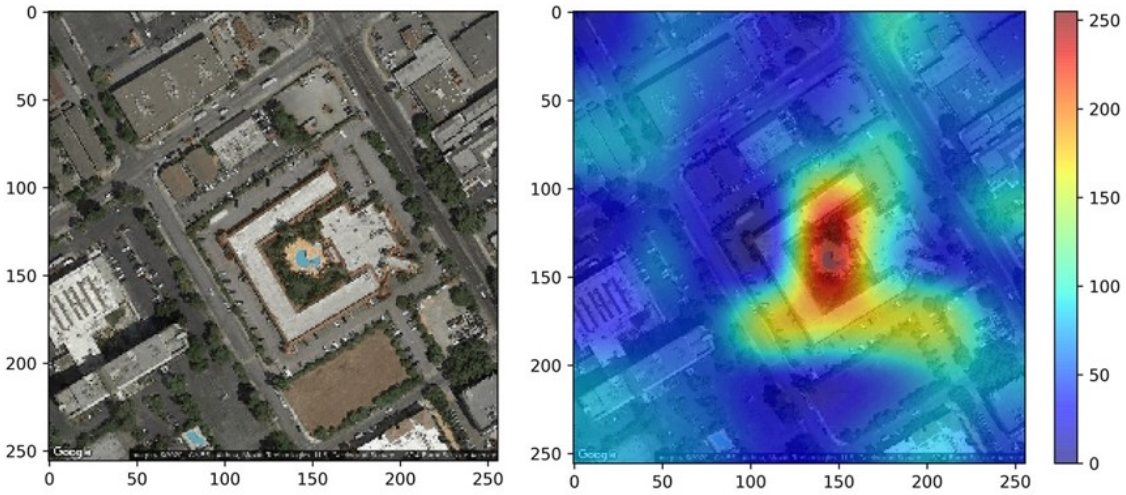

Figure 29: A false positive example from the test dataset.

A.3 False Negatives Cases

False negative (FN), also known as type 2 error, is the case where the sample does contain a helipad, however the system predicts no helipad. The FNs are cases where the CNN failed, and insights for the reasons the CNN failed can be found from inspecting these cases. This case along with the TP case can be used to determine the recall of the system using the equation below. Recall represents the percent of times the system correctly classified images with a helipad.

$$Recall = \frac{TP}{TP + FN} \tag{6}$$

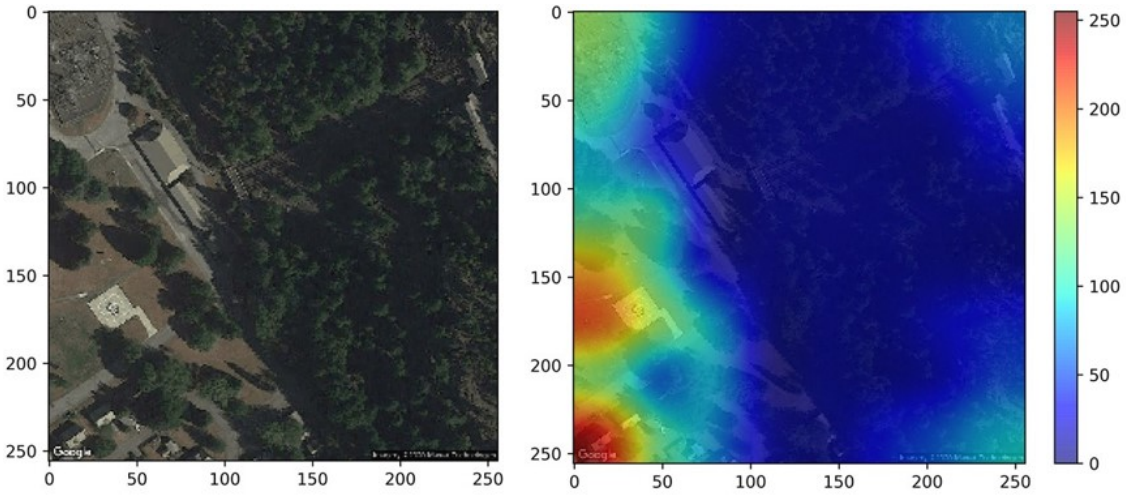

Figure 30: A false negative example from the test dataset.

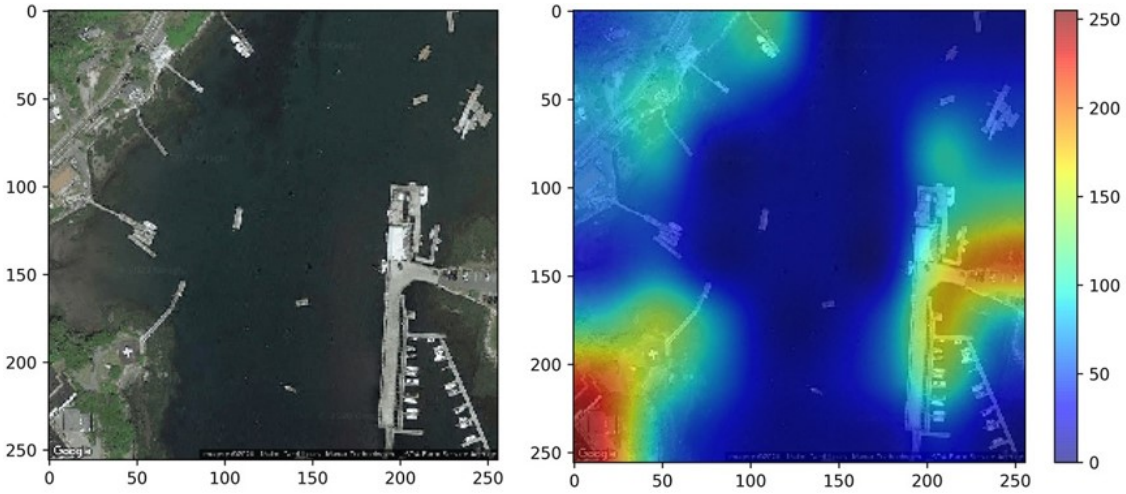

Figure 31: A false negative example from the test dataset.

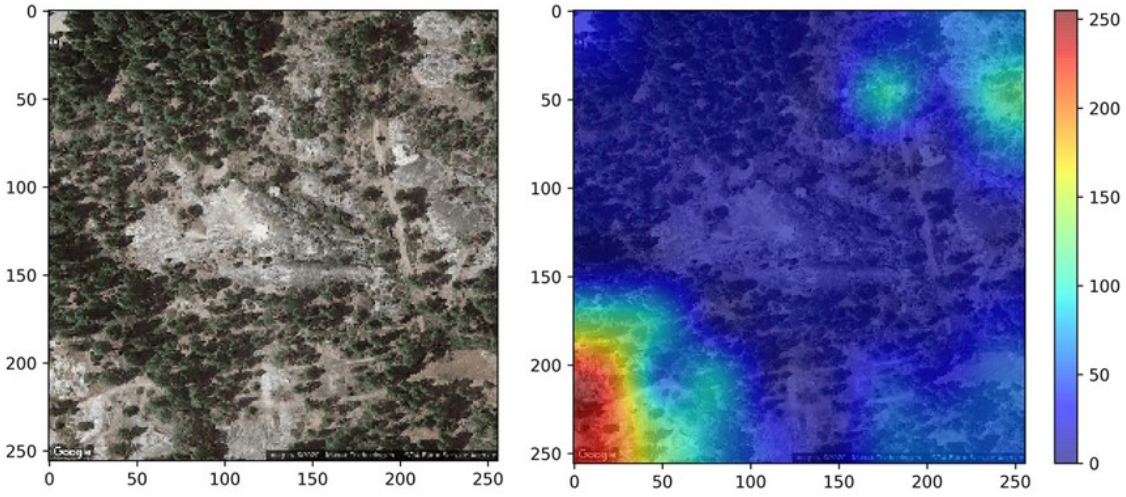

Figure 32: A false negative example from the test dataset.

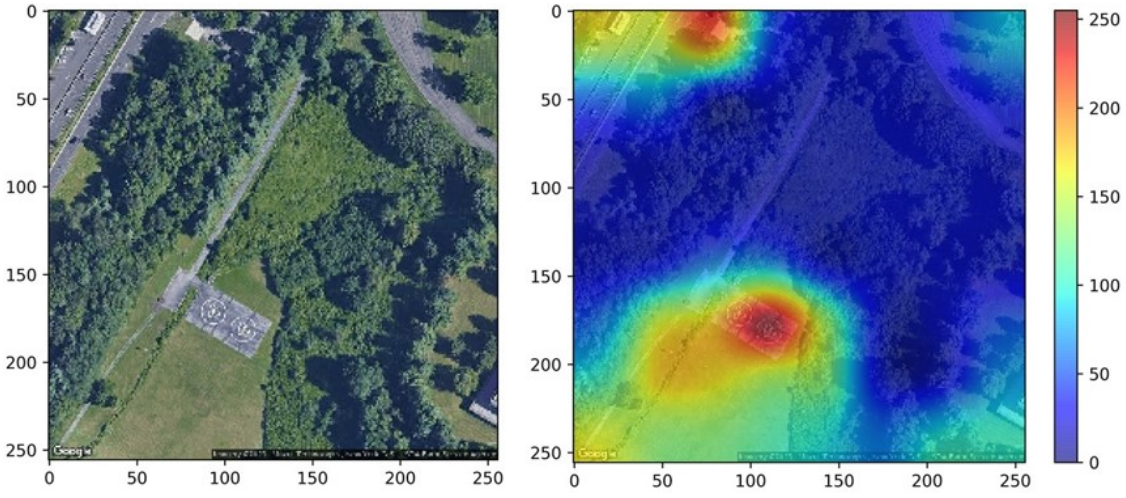

Figure 33: A false negative example from the test dataset.

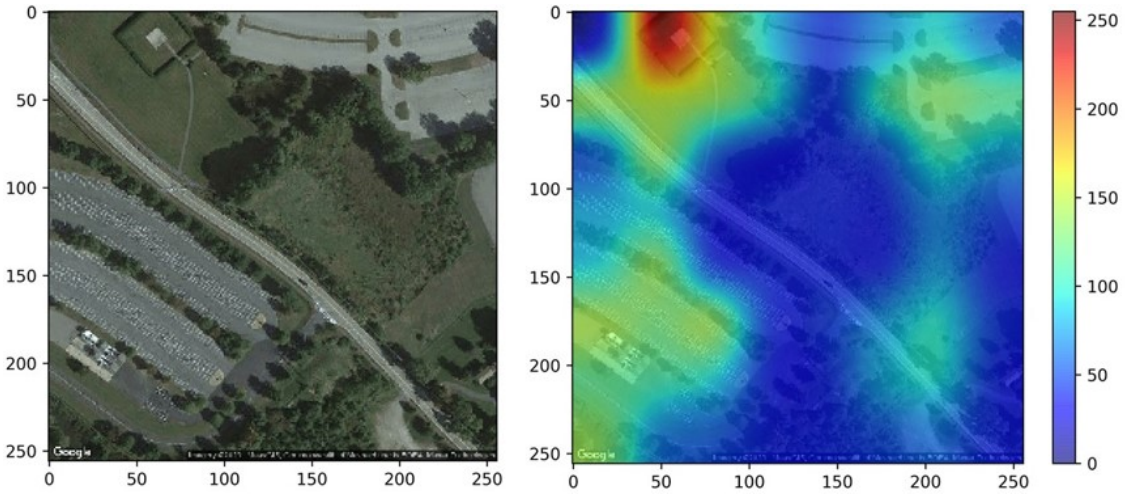

Figure 34: A false negative example from the test dataset.

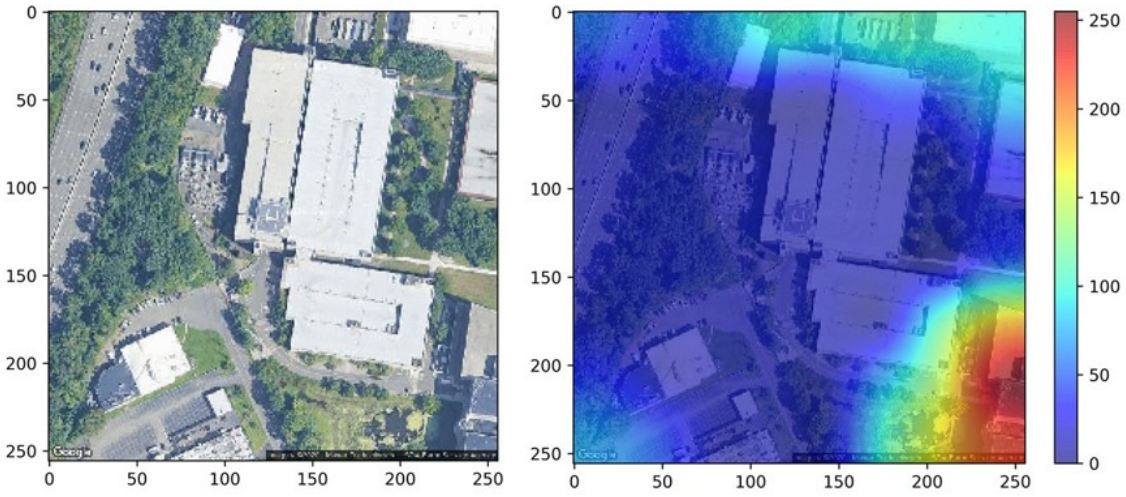

Figure 35: A false negative example from the test dataset.

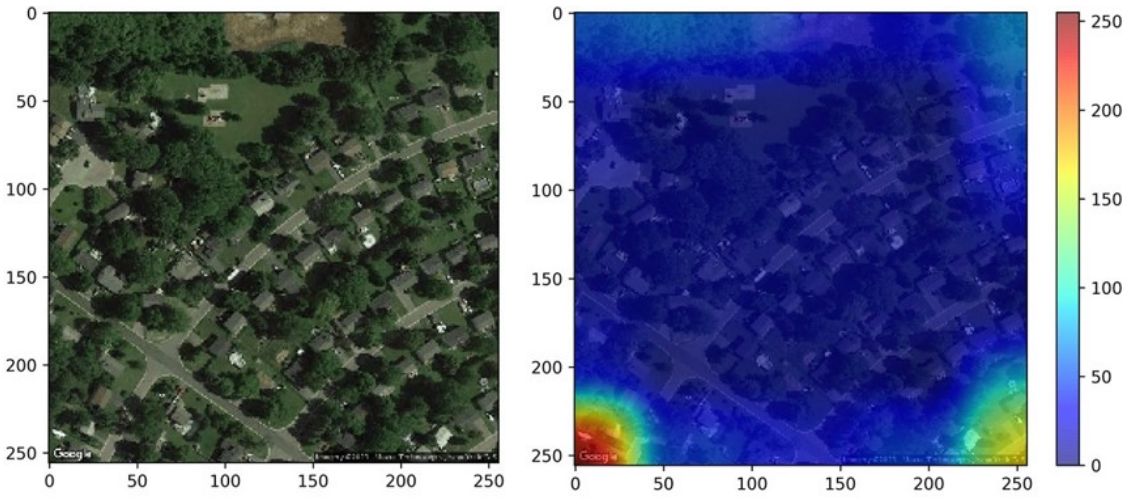

Figure 36: A false negative example from the test dataset.

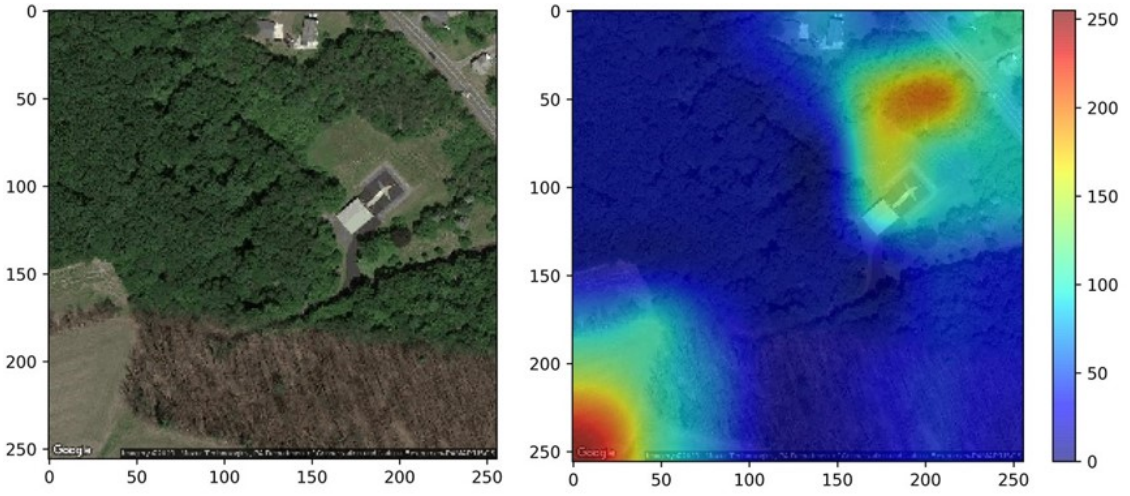

Figure 37: A false negative example from the test dataset.