HelipadCat: Categorised Helipad Image Dataset and Detection Method

Jonas Bitoun
School of Computing
National University of Singapore
Email: jonas@comp.nus.edu.sg

Stefan Winkler
School of Computing
National University of Singapore
Email: winkler@comp.nus.edu.sg

***Abstract*—We present HelipadCat, a dataset of aerial images of helipads, together with a method to identify and locate such helipads from the air. Based on the FAA's database of US airports, we create the first dataset of helipads, including a classification by visual helipad shape and features, which we make available to the research community. The dataset includes nearly 6,000 images with 12 different categories.**

We then train several Mask-RCNN models based on ResNet101 using our dataset. Image augmentation is applied according to learned augmentation policies. We characterize the performance of the models on HelipadCat and pick the best-performing configuration. We further evaluate that model on the metropolitan area of Manila and show that it is able to detect helipads successfully, with their exact geographical coordinates, in another country. To reduce false positives, the bounding boxes are filtered by confidence score, size, and the presence of shadows. Dataset and code are available for download.

I. Introduction

Urban Air Mobility (UAM) is expected to be integrated into the airspace [1] and expand quickly as an alternative to cars, bus or subway for inner-city or intercity commutes as well as for logistics services and deliveries. The increasing demand for these services, combined with the spread of helicopters, Unmanned Aerial Vehicles (UAVs) and autonomous drones entices the industry to make these services affordable to more people. For example, Uber has been offering trips from JFK airport to Lower Manhattan in private helicopters since July 2019. Also, startups partner with helicopter operators to offer ride-sharing platforms proposing time-saving trips by air in high-traffic areas, like Ascent Flights in the Philippines. Such co-operations allow cheaper flights and better use of available helicopters. In addition, several companies are developing their own eVTOL (electrical Vertical Take Off and Landing) aircraft [2] for short-distance urban travel as the transition to electric motors reduces the cost of vertical flights [3].

No global helipad database is available at the moment, except in the United States. The Federal Aviation Administration (FAA) lists the country's helipad coordinates, but it is incomplete and out-of-date, as it relies on information supplied by the heliport facility owners. Automatically cataloging all heliports in a country could help overcome these issues and lead to a more complete database, for use by helicopter operators, transportation companies, and researchers.

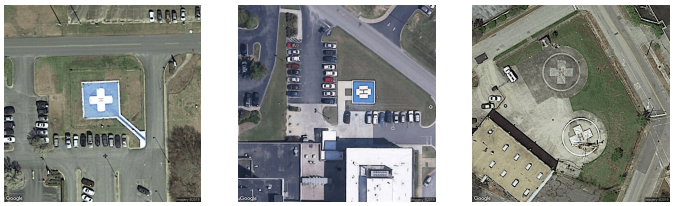
II. Related Work

The helipad detection problem was first addressed in [4], where a detection algorithm is designed using normalized wavelet descriptors for one specific pattern of helipad. [5] proposes a real-time detection method, with the purpose of allowing a UAV to land properly, by recognizing the international standard helipad pattern. Feature matching with Speeded-up Robust Features (SURF) [6] is used in [7] to detect one particular template of helipad in images. These algorithms are invariant to rotation and scale change, but are limited to one specific pattern. Besides, they are primarily designed to be used directly from a helicopter or an UAV for an imminent landing. They are also not database-driven.

[8] applies Deep Learning to the problem, by training a CNN model based on ResNet50 [9], with Transfer Learning on the final layer, using a sliding window approach to object detection. Because this approach generates many false positives or recognizes the same helipad more than once, DBSCAN clustering algorithm [10] is applied to filter the results. The number of false positives is still too high to scale the detection to entire cities. Our contribution tackles the issue by building an accurate dataset with ground truth and categorizing the different patterns of helipads found. Based on ResNet101 and trained on more layers, we aim to improve the model's precision. We further reduce false positive detections by applying shadow detection, since helipads are flat and free of shadows, and by filtering bounding boxes with ground areas too small or too large to fit a helipad. We make the HelipadCat dataset and code available for download.[1]

III. HelipadCat Dataset

No explicit dataset of helipad images is available. In the United States, the FAA publishes a database containing the coordinates of the country's airports and helipads [11]. Based on this information, we retrieve the corresponding aerial images using the Google Maps API [12] with the respective helipad coordinates, which returns an image tile with a resolution of 640×640 pixels.



(a) Centered (b) Small (c) Multiple

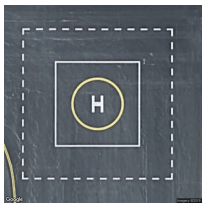
Fig. 1: Examples of helipad images in the database.

Out of the 5,878 images retrieved, only 3,264 (55.5%) have one or more helipads inside, for a total of 3,463 helipads.

[1] https://github.com/jonasbtn/helipad_detection

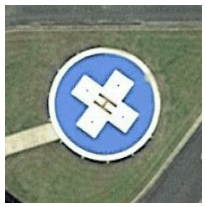
Besides, the size of the helipads varies significantly, and about half are not centered in the image, meaning that their registered coordinates are not precise. Although this database is not perfect, it is relatively balanced between true and false samples. Figure 1 shows sample images from the dataset.

Ground truth for each image is created manually, using a custom-made interface. We annotate the images by drawing a bounding box around the helipad when there is one. Moreover, a category is assigned to each pattern of helipads found, for a total of 12 distinct categories. Figure 2 shows examples of the pattern of each group. Note that the categories' sizes are unbalanced, and the number of possible patterns is quite high.



(0) Count: 60

(1) Count: 1616

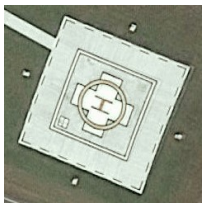


(2) Count: 447

(3) Count: 142

(4) Count: 241

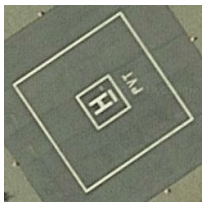
(5) Count: 113



(6) Count: 141

(7) Count: 89

(8) Count: 143



(9) Count: 207

(Others) Count: 64

(Unknown) Count: 29

Fig. 2: Helipad categories with their respective sample counts.

74% of the dataset (2,418 helipads) are used for training, while the remaining 26% (846 helipads) are set aside for testing.

In order to make HelipadCat easily shareable, we put all the metadata inside a separate file for each image.[2] This JSON file contains the helipad coordinates, the image zoom level, the index inside the FAA database, the Google Maps URL to re-download it, and the ground truth annotation (including category and bounding boxes).

[2] Available at https://github.com/jonasbtn/helipad_detection

IV. Dataset Augmentation

With relatively few examples of helipads, image augmentation is necessary. A team at Google recently conducted an experiment of learning the optimal augmentation policy for object detection using reinforcement learning [13]. They found that learned augmentation is beneficial for small datasets and small objects, which fits exactly our case. We use their latest policy to generate augmented images from our training set. It includes many sub-policies applied sequentially on the image with a certain probability (*Posterize*, *Sharpness*, *Equalize*, *Autocontrast*, *Solarize* and *Brightness*).

We establish the following augmentation strategies:

- Strategy 1: Generate one augmented image per example in the training set.
- Strategy 2: Balance the number of augmented images per categories by augmenting them a number of times to match the size of the largest category.
- Strategy 3: Duplicate each category with augmentation a specific number of times to reduce false positive and false negative.

Since each sub-policy is applied with a probability, we end up having different augmentations for the same example. The number of augmented images generated by each strategy is listed in Table I.

TABLE I: Categories' size after applying augmentation strategies. The training set is composed of the original dataset and the images generated by one augmentation strategy.

Category	Original	S1	S2	S3
0	40	40	1,161	80
1	1,161	1,161	1,161	2,322
2	309	309	1,161	618
3	118	118	1,161	708
4	165	165	1,161	165
5	98	98	1,161	294
6	85	85	1,161	340
7	68	68	1,161	272
8	115	115	1,161	575
9	172	172	1,161	860
Others	58	58	1,161	58
Unknown	29	29	1,161	58
Total	2,418	2,418	13,932	6,466

V. Model

We train a Mask R-CNN with a ResNet101 [14] backbone to generate the bounding boxes for each instance of an object in an image. This neural network is composed of 101 layers with skip-connections. The trained model size is 250MB. In order to speed up the training, we initiate the model with the pre-trained weights of MS COCO [15]. We use the API developed in [16], combined with our code, to load our dataset, train the model, and predict the helipads. The API allows us to choose which layer to train, and to freeze the weights of the others. Starting by training only the head layers, we gradually increase the number of layers to train. The training configurations and parameters of our different models are given in Table II. Note that more layers slow down the training. The smooth-L1 loss, the objective, converges, as shown in Figure 3.

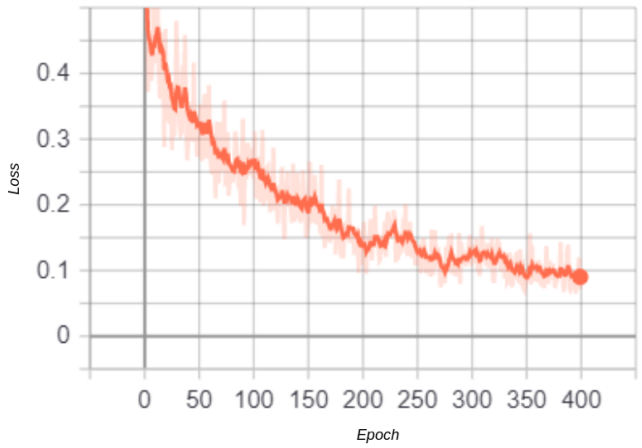


Fig. 3: Smooth-L1 loss curve of model D.

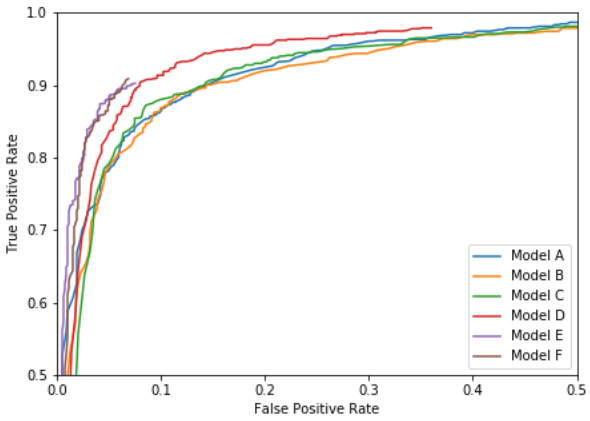


Fig. 4: ROC curves for the test set (view in color).

The output of the model is a bounding box together with a confidence score. Detected bounding boxes can be filtered using rule-based strategies, developed and fine-tuned according to the observations made from our results:

1) If one bounding box is contained inside another, only the one with the highest score is kept.
2) If two bounding boxes overlap by more than 50%, only the one with the highest score is kept.
3) A bounding box is kept if its score is above a certain threshold.

VI. Results

During training, the current weights are saved after each epoch. After training, choosing the right epoch is essential to prevent under- and overfitting. We keep the weights from the epoch where both the training and validation loss are minimal. Then, we evaluate the model for both the training and test set using Mean Average Precision (mAP), computed using the API of [16]. Different models are trained with distinct configurations. The results are shown in Table II.

TABLE II: Training parameters and results.

Model	A	B	C	D	E	F
Layers	heads	heads	heads	5+	3+	3+
Aug	S1	S2	S3	S2	S2	S2
Best Epoch	88	209	228	381	288	257
Loss	0.43	0.29	0.35	0.08	0.05	0.04
Val Loss	0.41	0.33	0.37	0.27	0.17	0.76
Train mAP	0.91	0.92	0.93	0.97	0.96	0.95
Test mAP	0.92	0.92	0.93	0.93	0.87	0.88

We also evaluate Accuracy, Error, Precision and Recall for different score thresholds. Figure 4 shows the receiver operating characteristic (ROC) curves of each model, which help us visualize the performance of each model. The AUC is defined as the area below the ROC curve. The closer the AUC is to 1, the better the model's ability to detect a helipad. Models A–C, with only the head layers trained, perform similarly, even though the augmentation strategies and training configurations are different. Models D–F, where layers from stage 5 and above are included in the training, perform better.

All evaluation metrics are listed in Table III. Even though model D achieves a higher True Positive rate compared to model E, without score filtering (98% vs. 90%), its False Positive Rate is high (36% vs. 7.5%). Since we want to minimize false positives for our given application (avoiding to land on a site without helipad), model E appears to be the best model (after filtering all the bounding boxes with score below 0.95).

TABLE III: Evaluation metrics on the test set.

Model	D	E	E	F
Score Filtering	0	0	0.95	0.95
True Positive Rate	0.98	0.90	0.87	0.89
False Positive Rate	0.36	0.075	0.04	0.04
Accuracy	0.80	0.91	0.91	0.92
Error	0.19	0.08	0.09	0.07
Precision	0.72	0.92	0.96	0.96
Recall	0.98	0.90	0.87	0.89

We compare our method with the another CNN-based approach [8]. Although the test areas are different, it can serve as a rough baseline. [8] achieves a precision of 67.2% and a recall of 90.0% on its test set, while our model F reach a precision 96% and a recall of 89.0% with score filtering. This large improvement in precision may be the result of the method enhancement. Indeed, we upgraded the dataset by adding pattern categorization and image augmentation. Furthermore, our model was trained on all layers of Resnet101, whereas [8] trained its model with transfer learning on the final layer of Resnet50.

VII. Helipad Discovery

Our goal is to build a global helipad database with precise geographical coordinates for flight planning. We would like to find helipads in unknown territory for mapping purposes.

To explore the model behavior at scale, additional unseen data are gathered using satellite images from the greater Manila region. The city is known for being home to a high helicopter activity and thus numerous helipads. However, we expect significant differences between the training dataset (consisting of US helipads) and the new data from a different

country. For our scenario, we download the entire city from Google Maps at a zoom level of 19. This results in a new (unlabeled) dataset of 263,496 images, each of dimension 256x256.

A. Model/Training Improvements

After running the predictions with model E, we manually check all the helipad detections. We notice a substantial number of false positives. Indeed, regular shapes like squares and circles are detected as helipads, especially the roofs of houses and buildings, sometimes with a very high confidence score. We suspect that categories 4 and 7 may be causing this problem. Therefore, we train model F without the images from those categories, coupled with augmentation strategy 2. This produces better results thanks to the larger dataset size and reduced exposure to examples resembling false positive. The results of both models are included in Tables II and III.

According to our results, removing categories 4 and 7 has a positive impact on the benchmark. Indeed, after applying the filters using the same score threshold, the accuracy and recall have increased by 1% and 2% respectively, and the error reduced by 2%. While the false positive rate stays the same, the true positive rate has increased by 2%.

Subsequently, we train two new models denoted as F' and G on all layers of Resnet101, with additional improvements designed to reduce false positives as much as possible. First, we remove all images with generic circle and square shapes from the training set. Specifically, categories 1, 2, 3, 5, 6, 8, and 9 now constitute the training set, while the remaining categories 4, 7, Others and Unknown are moved to the validation set. The goal is to have low performance on the validation set so that generic squares and circles are ignored in the training.

We also develop our own augmentation policy using the *ImgAug* package [17], to fix some of the apparent weaknesses of Google's approach from [13]. Our policy includes many sub-policies (*FlipLR*, *FlipUD*, *Rotate*, *Affine rescale*, *GaussianBlur*, *HistogramEqualization*, *ShearX*, *ShearY*, *EnhanceSharpness* and *EnhanceBrightness*) applied sequentially on the image with a certain probability.

Model F' is trained using Google's policy, whereas model G is trained with our own augmentation policy. As for model F, augmentation strategy 2 is used to train model F'. We adopt strategy 3 (see Section IV) for model G by duplicating each category 15 times to generate a very big dataset.

B. Additional Filters

1) Filtering Shadows: In order for a helicopter to land safely, helipads need to be on a flat surface, without nearby structures, hence free of any shadows. Shadows are evidence of 3D structures that might impede a helicopter landing and reduce the chances of a practicable helipad within the detected bounding box. Some examples of shadows around false positive helipad detections from the Manila dataset are shown in Figure 5. Therefore we implement the shadow detection algorithm described in [18]. In the first pass, this algorithm selects shadow seeds from pixels satisfying specific properties. In the second pass, region growing spreads the shadow from the seeds to cover the entire shadow area. For our application, this second pass is not needed.

We expand the bounding boxes slightly (by five pixels) in each direction to cover the surroundings of the helipad candidate. A bounding box is considered to have shadows if it has more than 3 seeds after the first pass of the algorithm, in which case we do not consider it a helipad.

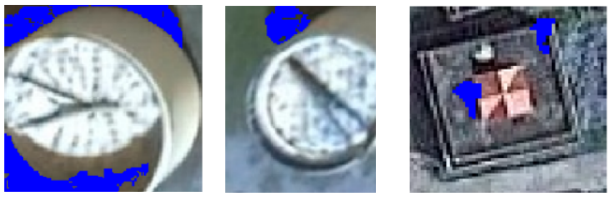


Fig. 5: Examples of shadows detected in false positives

2) Filtering by Area: Helipads need a certain area to provide enough space for a helicopter to land. According to the FAA's Advisory Circular on Heliport Design [19], the minimum width, length or diameter of the final approach and takeoff area (FATO) of a general aviation helipad is 1.5 times the overall length of the helicopter. Among the most popular commercial helicopters, we find the Eurocopter AS350 with a length of 11 meters. An ideal landing area for this model would be at least $\pi(11/2)^2 = 213m^2$ or $(11 \times 1.5)^2 = 272m^2$. In other words, a landing area of $300m^2$ would comfortably fit most light helicopters.

After obtaining the GPS coordinates of the four corners of each bounding boxes, the physical length and width are computed. To account the fact that bounding boxes are generally a bit larger than the actual helipad, and to cater to a range of helicopter sizes with lengths of 10-15m, we accept bounding boxes with an area between $160m^2$ and $550m^2$ and filter all other candidates.

3) Filtering by Score: This last filter removes all remaining bounding boxes below a certain confidence score. This threshold is for the user to choose and depends on the application. A higher threshold will reduce false (and some true) positives, whereas a lower threshold will increase them.

C. Results

The model is run to detect helipads on the Manila data. We manually classify the detected bounding boxes into true and false positives in order to evaluate the performance of the model with our filters. The results are presented in Table IV.

TABLE IV: Results of models F' and G on the Manila data, with the filters (shadow, area, score) activated in sequence.

Model	F'				G			
Shadow	No	Yes	Yes	Yes	No	Yes	Yes	Yes
Area	No	No	Yes	Yes	No	No	Yes	Yes
Score	0	0	0	0.99	0	0	0	0.99
Detections	1941				607			
TP	129	100	87	69	66	51	45	20
TN	0	564	1111	1590	0	208	348	511
FP	1812	1248	701	222	541	333	193	30
FN	0	29	42	60	0	15	21	46
Accuracy	0.06	0.34	0.61	0.85	0.10	0.42	0.65	0.87
Error	0.93	0.65	0.38	0.14	0.89	0.57	0.35	0.12
Precision	0.06	0.07	0.11	0.23	0.10	0.13	0.19	0.4
Recall	1	0.77	0.67	0.53	1	0.77	0.68	0.3

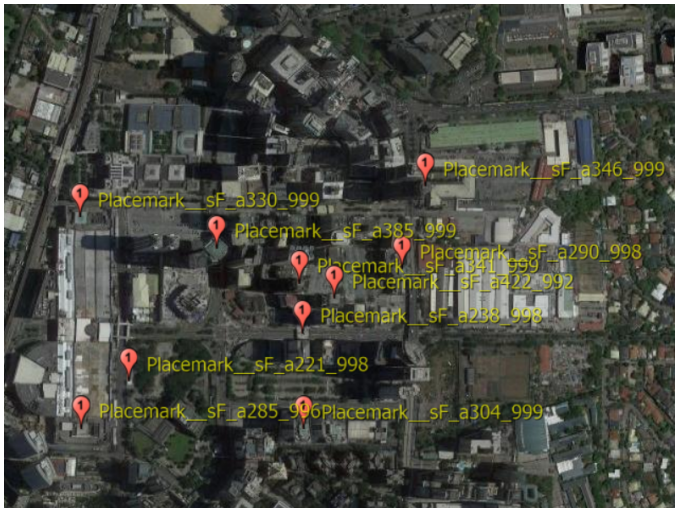


Fig. 6: Helipads detected by Model F' on the map with precise GPS coordinates.

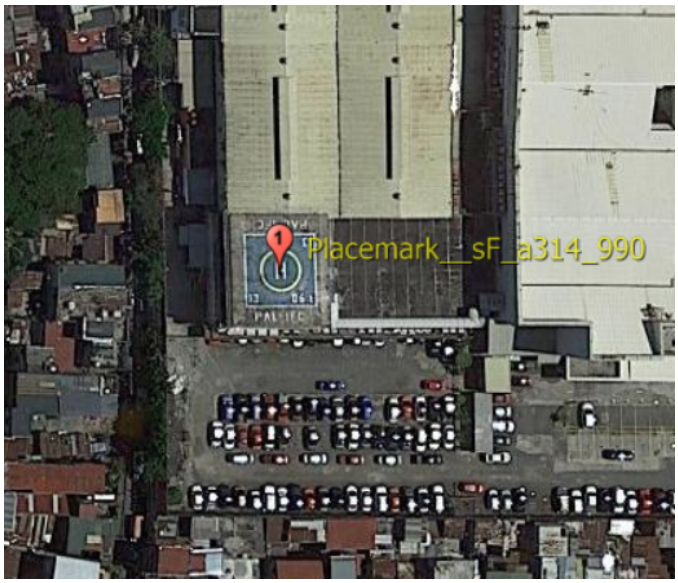


Fig. 7: Closeup of a helipad detected by Model F'. The yellow label indicates the properties of the helipad: no shadow detected (*sF* as *Shadow False*), ground area of $314m^2$ (*a314*), and a confidence score of 99% (*990*).

The effectiveness of the filters is evident from the results. The shadow filter successfully removes 564 and 208 false positives for models F' and G, respectively, while losing only 29 and 15 true positives. The area filter removes a further 547 and 140 false positives, while losing only 13 and 6 true positives. The score filter (with a threshold of 0.99) removes 479 and 163 false positives while losing only 18 and 25 true positives. Figures 6 and 7 visualize the detected helipads on the satellite map.

VIII. Conclusions

We created HelipadCat, a dataset of helipad images with ground truth including geolocation, helipad categories, bounding boxes, and exploring different augmentation strategies. Training more layers has a positive impact on the benchmark metrics. We obtain good models for helipad detection. The detected bounding boxes are filtered using shadow detection, area and score thresholds. Our three custom filters are effective, increasing model accuracy and reducing false positives. The city of Manila is analyzed extensively where many helipads are located. However, this particular city is not representative of the entire world. This is why we would like to extend the study to other regions.

The ultimate goal is to run the prediction on entire countries to construct a global helipad database. With the support of Ascent Flights, we plan to verify the certification of detected helipads, as the feasibility of a helicopter landing requires compliance with rules imposed by local regulations. Furthermore, we plan to expand the detection from designated helipads to any area suitable for landing a helicopter or heavy-lifting drone in urban areas.

References

[1] D. P. Thipphavong, R. Apaza, B. Barmore, V. Battiste, B. Burian, Q. Dao, M. Feary, S. Go, K. H. Goodrich, J. Homola *et al.*, "Urban air mobility airspace integration concepts and considerations," in *Proc. Aviation Technology, Integration, and Operations (ATIO) Conference*, 2018, p. 3676.

[2] N. Polaczyk, E. Trombino, P. Wei, and M. Mitici, "A review of current technology and research in urban on-demand air mobility applications," in *Proc. 8th Biennial Autonomous VTOL Technical Meeting and 6th Annual Electric VTOL Symposium*, 2019.

[3] M. J. Duffy, S. R. Wakayama, and R. Hupp, "A study in reducing the cost of vertical flight with electric propulsion," in *Proc. AIAA Aviation Technology, Integration, and Operations (ATIO) Conference*, 2017, p. 3442.

[4] G. F. Nsogo, K. Kith, B. J. van Wyk, and M. A. van Wyk, "Robust helipad detection algorithm," in *Proc. AFRICON*, Sept. 2007, pp. 1–7.

[5] S. Lee, J. Jang, and K. Baek, "Implementation of vision-based real time helipad detection system," in *Proc. 12th International Conference on Control, Automation and Systems (ICCAS)*, Oct. 2012, pp. 191–194.

[6] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," in *Proc. European Conference on Computer Vision (ECCV)*, 2006, pp. 404–417.

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

[8] G. Walker, "Detecting helipads using CNN," https://www.garethwalker.me/detecting-helipads-using-cnn, 2019.

[9] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Computer Vision and Pattern Recognition (CVPR)*, 2016.

[10] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proc. 2nd International Conference on Knowledge Discovery and Data Mining (KDD)*, vol. 96, no. 34, 1996, pp. 226–231.

[11] Federal Aviation Administration, "Airport facilities data," https://www.faa.gov/airports/airport_safety/airportdata_5010, 2020.

[12] Google Inc, "Google Maps API," https://cloud.google.com/maps-platform/maps, 2020.

[13] B. Zoph, E. D. Cubuk, G. Ghiasi, T.-Y. Lin, J. Shlens, and Q. V. Le, "Learning data augmentation strategies for object detection," in *Proc. IEEE Computer Vision and Pattern Recognition (CVPR)*, 2019.

[14] K. He, G. Gkioxari, P. Dollár, and R. B. Girshick, "Mask R-CNN," in *Proc. IEEE International Conference on Computer Vision (ICCV)*, 2017.

[15] T. Lin, M. Maire, S. J. Belongie, L. D. Bourdev, R. B. Girshick, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common objects in context," in *Proc. European Conference on Computer Vision (ECCV)*, 2014, pp. 740–755.

[16] W. Abdulla, "Mask R-CNN for object detection and instance segmentation on keras and tensorflow," https://github.com/matterport/Mask_RCNN, 2017.

[17] A. B. Jung, K. Wada, J. Crall, S. Tanaka, J. Graving, C. Reinders, S. Yadav, J. Banerjee, G. Vecsei, A. Kraft, Z. Rui, J. Borovec, C. Vallentin, S. Zhydenko, K. Pfeiffer, B. Cook, I. Fernández, F.-M. De Rainville, C.-H. Weng, A. Ayala-Acevedo, R. Meudec, M. Laporte *et al.*, "imgaug," https://github.com/aleju/imgaug, 2020, online; accessed 01-Feb-2020.

[18] V. Arévalo, J. Gonzélez, and G. Ambrosio, "Shadow detection in colour high-resolution satellite images," *International Journal of Remote Sensing*, vol. 29, no. 7, pp. 1945–1963, 2008.

[19] Federal Aviation Administration, "Heliport Design," AC 150/5390-2C, 2012.