









Applications of Classical and Deep Learning Techniques for Polar Bear Detection and Recognition from Aero Photography

Mikhail A. Nakhatovich^{1,2} , Ilya Y. Surikov^{1,2} , Vladimir Chernook³ ,
Natalia Chernook³ , and Daniil A. Savchuk^{1,2}  

¹ Peter the Great St. Petersburg Polytechnic University (SPbPU), Polytechnicheskaya 29,
195251 St. Petersburg, Russia

dsavchuk@itsociety.su

² ITSociety LTD, Diagonalnaya 4-1-170, 194100 St. Petersburg, Russia

³ Autonomous Non-Commercial Organization «Ecological Center «ECOFACOR», 11/1-10 N
Neyshlotskiy Lane, 194044 St. Petersburg, Russia

Abstract. The problem of detecting polar bears on the image taken from the plane is essential for ecologists who are tracking the disappearing population of the arctic inhabitants. The main challenge for this problem is to detect the white bear on the white ice. This paper covers the approaches which have shown valuable results for contrast objects captured from the plane, like cars, ships, and many others, instead of the polar bears that look blurry on the ice. However, the introduced approach based on both statistical and machine learning methods made it possible to build a tool that increases the semi-automatic bear detection rate a dozen times. The source data consists of 7360×4912 px aerial images, each image covering about the 21600 sq.m. of ice. On average, only one bear appears on every 1000 photos. The best-fit parameters for the solution gave a result of about 100% by recall metric and 51% by precision metric. The main strength of this solution is that it allows for finding almost all bears with a moderate amount of false-positive detections.

Keywords: Polar bear detection · Image processing · Deep machine learning · Transfer learning · Object detection · Augmentation

1 Introduction

A polar bear is at the top of the trophic pyramid in the Arctic and is an integral indicator of the state of the entire Arctic ecosystem. Therefore, regular instrumental monitoring of the polar bear population is outstandingly important. The main parameters of the population's state are the number and distribution of animals that can be obtained during regular aerial surveys. The main problem of detecting polar bears in aerial photography is the masking color and peculiar heat-protective fur, which makes these animals poorly visible in photos and IR images. After accounting flights, specialists have to process hundreds of thousands of aerial photographs manually, and the processing lasts for years.

The distribution density of polar bears in the spring on ice is low, and, as a rule, just one bear (sometimes with cubs) appears in several thousand of the processed photographs. Given such a low density, it is critical not to miss a single picture with polar bears.

The problem of detecting polar bears on images taken from a plane is an object detection task in aerial images. The detection of an object in aerial photography is complicated since objects in aerial photographs are tiny. Also, the light conditions such as the angle of observation, camera settings, fog, and daylight conditions are essentially affecting factors that could be detected in aerial photography. Finally, the main difficulty of the task is that the color of the object almost matches the background color.

The automatic recognition system (ROI generation system) is performed in this work and based on a combination of statistical methods in a row with machine learning, which allowed to achieve high accuracy with a small training dataset. An additional complication of the task was that no bear should have been lost, so machine learning models must have about 100% recall metric.

The main objective of this paper is to describe an approach that uses the combination of classic computer vision methods and deep learning to enhance the overall result.

1.1 Previous Work

The existing solutions that provide the automatic location of objects in aerial photographs can be divided into three categories, which are distinguished by the use of deep learning. Methods from the first category [1–6] are based on statistical computer vision methods and basic machine learning algorithms. The main idea is to get the segmentation of possible candidates, and then clarify the result (refine the boundary of the desired object) using classic computer vision methods. For example, in work [2], mean-shift clustering is used for segmentation and a circle-frequency filter for filtering. The problem with such approaches is that they show low accuracy and hardly ever used in their pure form today.

The second category of methods uses deep learning. Deep learning has shown remarkable successes in image recognition in recent years. Convolutional neural networks (CNNs) have been successfully applied in the field of object detection in aerial photographs [7–10, 12]. The resolution of aerial photographs can reach 10000×10000 pixels, which prevents the use of a neural network for the entire image at once. Therefore, for sufficiently large images, sliding windows are used to obtain lower resolution images without loss of information. Object detection architectures can be divided into two categories: single-stage and two-stage. The difference is that two-stages architectures (Region-based) firstly classify potential objects into two classes: foreground and background. Hence, such models have lower performance but higher scores [11]. In paper [12] Faster R-CNN [13] and Yolo [14] architectures are compared on a car detection from aerial photographs task. A modified version of Faster R-CNN was applied in [8] for detecting vehicles. The rotation-invariant CNN model was introduced in [9]. An additional rotation-invariant layer applied to AlexNet made it better in the performance of object detection.

Finally, the third group of methods uses classic computer vision algorithms in a row with deep learning models, namely CNN. There are two main ideas across this group. The first idea is to extract the feature map from the image using CNN and then, based on

the obtained map, define object class using classic machine learning algorithms. In paper [15], AlexNet is used for feature extraction and SVM for classification. The second idea is quite similar to the idea of the statistical methods, but CNN is used to clarify if the object belongs to the desired class. In this paper, statistical analysis is used to propose regions of interest (ROIs), and CNN is used as the classifier of the proposed region.

2 The Input Data

The aerial survey of polar bears was carried out in the Russian part of the Chukchi Sea from April 18 to May 18, 2016. The survey was performed from a flight altitude of 250 m, during the daytime, mainly between 10- and 17-h local time. The dataset consists of 30 thousand high-resolution (7360×4912 px) photographs of the iced sea surface. Each image covers an area of 180×120 m. Illumination of the ice surface changed depending on the height of the sun and on the weather (sun, cloudiness, haze, fog). The example of the input image is presented in Fig. 1.



Fig. 1. A quarter of the input data example. A bear is only 50×50 px.

Since the density of the bears is usually low (one bear per 30–70 km of flight), 30 photographs with polar bears detected during the preliminary manual data processing were used for the train the neural network.

3 General Approach

The pipeline for image processing consists of several steps was developed. The main requirement for the pipeline was to add new steps easily or modifying existing ones. A scheme of the pipeline is presented in Fig. 2.

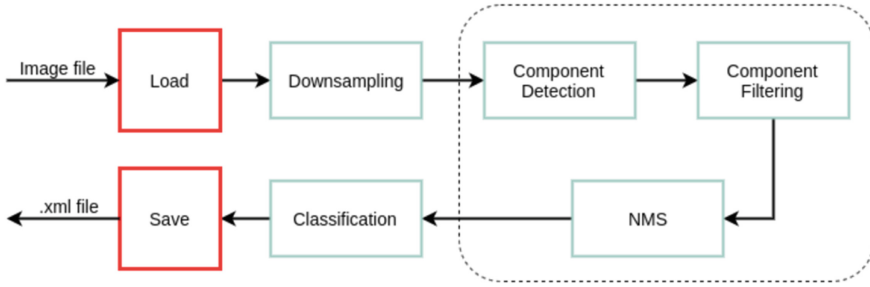


Fig. 2. Image processing pipeline.

In the first step, the image is compressed to a resolution two times smaller than the original. The compression allows for increasing the speed of image processing without losing important information.

Bears stand out against the background by the hue value. Hence, it is possible to empirically determine the range according to the hue channel in the HSV color space, which contains the color value of the bears. Figure 3 shows an example in which the bear is separated from the background using the hue channel.

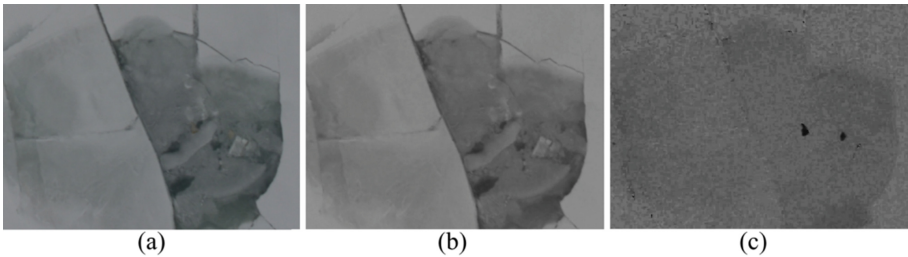


Fig. 3. (a) RGB image; (b) HSV Value channel image; (c) HSV Hue channel

On some images, some conditions make it difficult to determine a bear only by the color of its coat. For example, the sun shines from the back of the bear; the bear is in a fog. The fur visually turns gray, but it still contains useful information. Using standard effects to change images allows for enhancing this information. Therefore, two filters are applied to the original image:

1. “Light” filter: increased gamma correction, contrast, brightness, saturation, light, and sharpen.
2. “Sharp” filter: increased saturation and sharpen.

For each of the filters, a range on the hue channel can be determined. The examples are provided in Fig. 4. In images with the bear heavily lit by the sun, the snow around the bear also acquires a yellowish tint, which interferes with the correct segmentation by color. To avoid this, in addition to the hue range, the value range filter in the HSV color space is applied.

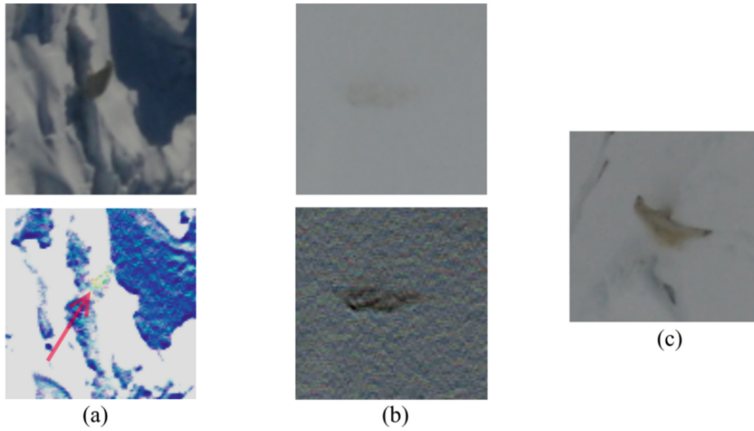


Fig. 4. Examples of images with bad weather conditions and effects allowing to highlight bear: (a) bear in shadow – “light” filter; (b) bear in fog – “sharp” filter; (c) normal image – default params

The result of segmentation is a map of ROIs – a black-and-white image on which areas of potential bears are marked in white (Fig. 6(b)).

The connected components of the image are found using the result of the previous step. The next step is to filter the found areas. Since the flight altitude is constant, the size of the bears does not significantly differ in aerial photographs. Therefore, the filtering of connected components is based on their size and shape. While components processing, next filters are applied to recline noises:

- component area;
- an aspect ratio of the rectangle built around component;
- component occupancy inside rectangle;

Adaptive Gaussian thresholding [16] is performed inside each rectangle on the grayscale, hue, and blue channels (Fig. 5). Next, for each channel, the percentage of black inside the rectangle is checked. If the value is out of an empirically selected range, it is discarded as noise. Thus, such noises as part of a block of snow or a highly illuminated surface are reclined. Figure 6 (c) illustrates the result of the component filtration.

The filtering result is a set of ROIs, which are the possible locations of bears. After filtering, detection results on three images (original and with filters) are combined, and the non-maximum suppression (NMS) technique is applied to unite the intersecting regions.

However, there are landscapes with objects that cannot be distinguished from bears based on the previously mentioned signs. A CNN-based classifier is used to eliminate them.

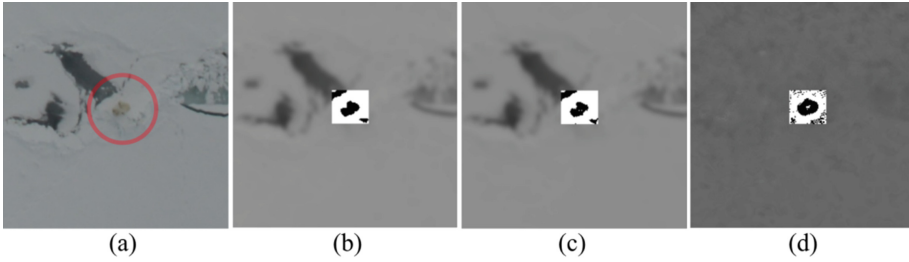


Fig. 5. Adaptive binarization: (a) source image; (b) blue channel; (c) grayscale channel; (d) hue channel.

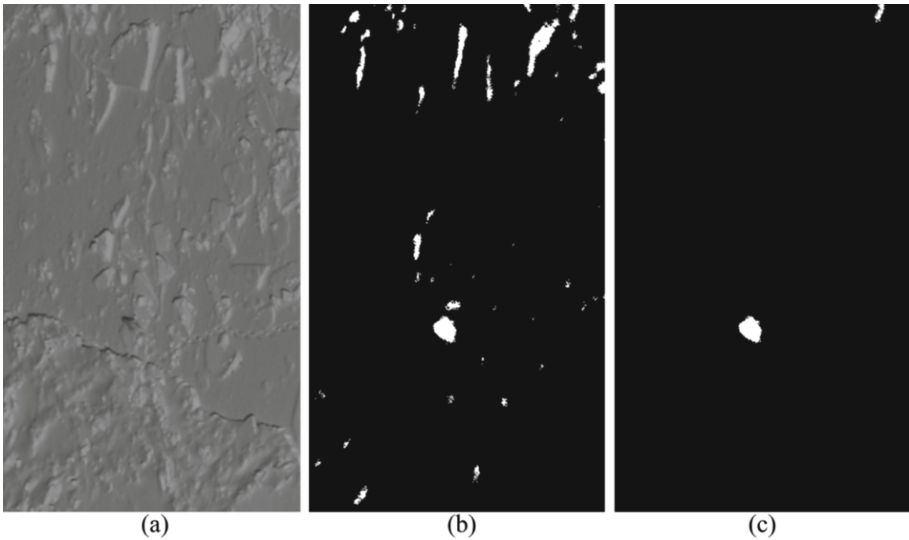


Fig. 6. Image processing stages: (a) source image; (b) ROIs after component detection; (c) component filtration result

3.1 Dataset

From the original image, 128×128 squares were cut for training the classifier. Some of them contain bears, while others are the ice surface (noise). As clippings of noise were selected, those areas that were segmented as bears after applying statistical methods.

Since the number of bears in the pictures is tiny, we had to deal with imbalanced classes dataset. Moreover, the task is complicated by the fact that the recall rate for bears must be as higher as possible since each bear is crucial for estimating the population density.

Ordinary augmentation, such as rotations and flips was not enough to achieve the necessary results, so we have expanded our dataset using simple method: for each bear, several images were added to the dataset, in which the bear was in different parts of the image (in the center, at the corners), as well as all kinds of rotations of these images. An

example of various instances presented in the dataset for one bear is shown in Fig. 7. Such a method allows increasing precision metric by 0.39 compared with ordinary augmentation.



Fig. 7. Augmentation example.

Thereby, a training classification dataset consists of 4320 images: 2160 of them are augmented bears, other – noises of various types.

The test dataset for classification contains 1152 images, of which 576 are augmented 8 test bears. The best results are presented as the best shot from cross-validation over all possible permutations of given bears with 8 bears taken as a validation base set. The cross-validation showed results are closely distributed from one to another run with general fluctuation, not more than 3% in Recall and 15% in Precision.

3.2 Classification

ResNet. The first method of bears classification, which was tried in the course of research is Residual Network (ResNet) [17] – a state-of-the-art neural network used

as a backbone for many computer vision tasks. It is pre-trained on the ImageNet dataset, so training this model even on an own small dataset can lead to good results. The output layer of this network was replaced to perform the task of one-class classification.

ResNet-34 and ResNet-50 were trained and tested. Other variations (18 layers, 101 layers) have shown worse results. The results of classification on train and test dataset are presented in Table 1. Models have been trained for 100 epochs; the best result was chosen from each training.

As can be seen from Table 1, ResNet-50 showed better results. Using this method allows throwing away most of the noise, while not losing a single bear.

Table 1. Comparison of using ResNet architectures.

	Recall	Precision
ResNet-34	1	0.273
ResNet-50	1	0.419

ResNet + SVM. In addition to the use of ResNet as the classifier, it can be used to extract features from the image. Then classic machine learning algorithms can be applied to classify objects using image features. One of these algorithms is a Support Vector Machine algorithm [18, 19] with the regularization parameter C, which compromises between correctness on a training set and maximization of the decision function’s margin. This algorithm was chosen as one of the classic methods which are effective in high dimensional spaces. The params for SVM were chosen empirically: RBF kernel, $\gamma = 1/n_features$, $C = 1.0$.

Table 2 shows the comparison between ResNet pre-trained on ImageNet and pre-trained ResNet with additional training on the collected dataset.

Table 2. Comparison of using ResNet architectures with SVM.

	Recall	Precision
Pre-trained ResNet-34 + SVM	0.778	0.055
Pre-trained ResNet-50 + SVM	0.806	0.071
Trained ResNet-34 + SVM	1	0.424
Trained ResNet-50 + SVM	1	0.507

Faster R-CNN. The other way to classify bear on the image is to try to detect them using CNN. If the model finds a bear on the ROI, then it is classified as a bear. Pre-trained on ImageNet dataset Faster R-CNN was chosen as one of the most widely used state-of-the-art architecture, which shows high accuracy even when training on a small dataset.

Two training approaches have been tested: one-class object detection (only bears) and two-classes (bears and noise). It turned out that training on two-classes object detection performs better. The results are provided in Table 3.

Table 3. Comparison of using Faster R-CNN with different numbers of classes.

	Recall	Precision
Faster R-CNN one-class	1	0.234
Faster R-CNN two-class	1	0.456

2 VAE+SSIM. The idea of using several Variational Autoencoders (VAEs) for classification task was proposed in [20]. The main idea is to exploit the primary objective of VAE (generation) for classification. For each class, VAE is trained in an unsupervised way. Then to find out which of the classes the object belongs to, the image is passed through every trained VAE, and then the SSIM index is calculated between input and output. The classifier operation scheme is presented in Fig. 8. The class whose VAE showed the best generation result (the largest SSIM index) is the class of the object. The process repeats 5 times to improve the prediction, and then the class that is chosen more times is selected as a predicted class.

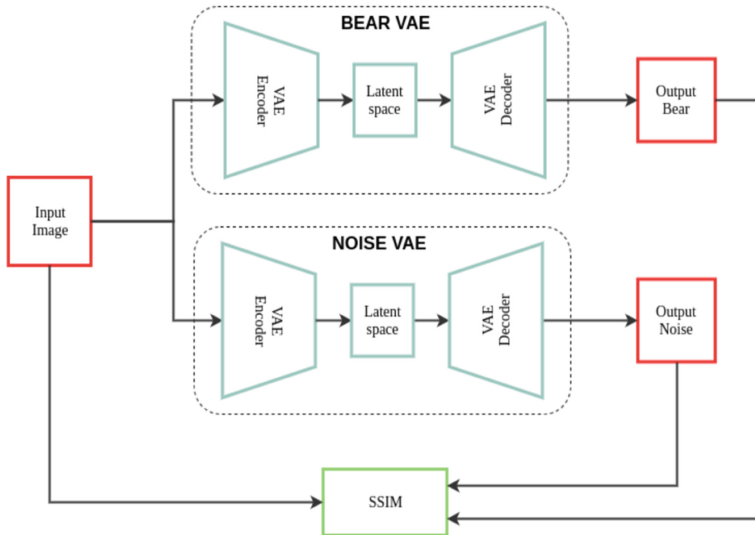


Fig. 8. 2 VAE approach scheme.

Two VAE were trained for polar bear detection task: one for bear generation, and one for noise. Figure 9 shows the autoencoders' results. It can be seen that the autoencoder

trained on bears generate output looks like a bear at the bear as the input while output from noise does not seem like a bear. However, not all the images are handled well. In most cases, the region with the noise is not so different from the region with the bear. So, the output from the VAE trained on bears is quite similar to the input noise.

The training metrics for this approach are presented in Table 5.

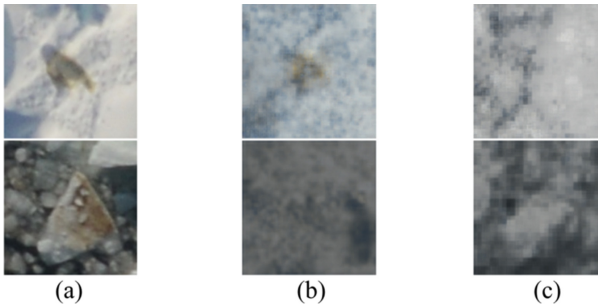


Fig. 9. Autoencoders results: (a) input region; (b) result of VAE trained on bears; (c) result of VAE trained on noises.

Thus, several approaches were tested for the classification task. ResNet-50 pre-trained on ImageNet and trained on the own dataset has shown the best result.

Sliding Window. Besides the general approach described above, a sliding window method was tested. The key idea is to put a ‘window’ on an image and run that part of the image through trained object detection CNN, slide the window and do it again. It is one of the most popular ways to process high-quality aerial images [8, 9]. Faster R-CNN with Inception v2 has been chosen as a model for object detection. However, this approach has shown worse results than the general approach.

Also, a modification with ResNet has been tried. Every predicted by Faster R-CNN bounding box is passed through a trained ResNet-50 classifier to reject some noises which have been recognized as bears. Statistical methods were applied to make up the dataset consist of noises of different types, which are similar in characteristics to bears.

The results of the sliding window approach are provided in Table 4.

Table 4. Comparison of sliding window methods results.

	Recall	Precision
Faster R-CNN	0.972	0.076
Faster R-CNN + ResNet50	0.972	0.271

4 Experiments

4.1 Performance

The developed approach has shown the ability to process the input data and find bears almost without lossless. Due to a large amount of data that needs to be processed, the performance of the trained model is an important issue while using an approach to investigate the ecological state during the plane expedition. The developed method should be able to process the 20 000 source images for 8 h on average using three laptops in parallel. 8 h is the time between possible arctic daytime flights. Table 5 provides the comparison of the mean image process time (including all steps of the pipeline). Computer params: Intel Core i5-8400, GeForce GTX 1080 Ti, and 16 GB RAM.

The system allows the assessor to look through the data ten times faster and more accurately when the exact number of bears needed. This statement has been tested with an independent group of 5 experts, who tried to find 3 bears on 100 photos with and without our approach that point assessor to the place in the image where bear possible could be. All experts have never seen the dataset even before.

4.2 Accuracy

Precision and Recall were chosen as the primary metrics for the validation of our approach. The Precision and Recall of a predicted set of bears' and a set of real bears' are calculated as:

$$Precision = \frac{TP}{TP + FP}; Recall = \frac{TP}{TP + FN} \quad (1)$$

where TP are the true positives, FP - false positives, and FN - false negatives. Precision shows how many selected items are relevant, and Recall shows how many relevant items are selected.

Table 5 shows a comparison of the best results of using different methods for the classification of ROIs and the mean computational time T.

Table 5. Comparison of classification best results.

	Recall	Precision	<T> , sec.
Without classification	1	0.015	3.81
ResNet-50	1	0.419	3.86
Trained ResNet-50 + SVM	1	0.507	3.90
Faster R-CNN two-class	1	0.456	5.81
2 VAE + SSIM	1	0.041	4.32
Sliding window (Faster R-CNN + ResNet50)	0.972	0.271	4.36

Analyzing Table 5, it was concluded that the ResNet-50 + SVM method shows the best ratio of accuracy and speed. However, other methods tested are also applicable

for solving the desired problem, although more regions with ice are detected as bears (Precision metric is lower). Table 6 provides the results for each step of the pipeline for the best method.

Table 6. Number of ROIs after each stage.

Stage	Number of ROIs
Component detection	1151849
Component filtering	2902
NMS	2425
Classification with the best method	71

5 Conclusions

The approach developed has shown the ability to be used as a reliable helper for counting and locating the polar bear appearances on aerial photography.

A lot of general techniques have been tried with valuable customizations developed. The best results with a 100% recall metric and a 51% precision metric have been shown by pipeline, including statistical ROI preparation step finalized with Trained ResNet-50 as feature extractor and SVM as classifier. The best approach shows the appropriate performance of 3.9 s on average for each photo, which makes it possible to use the method for performing calculations of 20 000 photos in 8 h (by three portable workstations) between flight sessions to collect statistics for correcting future flights.

The roadmap of this solution is to collect more data containing bears and incrementally increase the accuracy of the model and finally build a fully automatic tool for bear detection. To further improve the results on this task, it is recommended to speed up the ROI search algorithm. Most of the time, the system takes to process three images (2 filters and the original), so it would be better to find a filter that is universal for all cases, which allows you to process one image instead of three.

References

1. An, Z., Shi, Z., Teng, X., Yu, X., Tang, W.: An automated airplane detection system for large panchromatic image with high spatial resolution. *Optik – Int. J. Light Electron Opt.* **125**(12), 2768–2775 (2014)
2. Bo, S., Jing, Y.: Region-based airplane detection in remotely sensed imagery. In: 3rd International Congress on Image and Signal Processing, pp. 1923–1926 (2010)
3. Cai, H., Su, Y.: Airplane detection in remote sensing image with a circle-frequency filter. In: International Conference on Space Information Technology, p. 59852T (2005)
4. Moranduzzo, T., Melgani, F.: Automatic car counting method for unmanned aerial vehicle images. *IEEE Trans. Geosci. Remote Sens. (TGRS)* **52**(3), 1635–1647 (2014)

5. Yu, X., Shi, Z.: Vehicle detection in remote sensing imagery based on salient information and local shape feature. *Optik – Int. J. Light Electron Opt.* **126**(20), 2485–2490 (2015)
6. Shi, Z., Yu, X., Jiang, Z., Li, B.: Ship detection in high-resolution optical imagery based on anomaly detector and local shape feature. *IEEE Trans. Geosci. Remote Sens. (TGRS)* **52**(8), 4511–4523 (2014)
7. Li, K., Cheng, G., Bu, S., You, X.: Rotation-insensitive and context-augmented object detection in remote sensing images. *IEEE Trans. Geosci. Remote Sens. (TGRS)* **56**(4), 2337–2348 (2018)
8. Tang, T., Zhou, S., Deng, Z., Zou, H., Lei, L.: Vehicle detection in aerial images based on region convolutional neural networks and hard negative example mining. *Sensors (Basel, Switzerland)* **17**(2), 336 (2017)
9. Cheng, G., Zhou, P., Han, J.: Learning rotation-invariant convolutional neural networks for object detection in VHR optical remote sensing images. *IEEE Trans. Geosci. Remote Sens. (TGRS)* **54**(12), 7405–7415 (2016)
10. Tayara, H., Chong, K.T.: Object detection in very high-resolution aerial images using one-stage densely connected feature pyramid network. *Sensors (Basel, Switzerland)* **18**(10), 3341 (2018)
11. Huang, J., et al.: Speed/accuracy trade-offs for modern convolutional object detectors. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3296–3297 (2017)
12. Aerial Images Processing for Car Detection using Convolutional Neural Networks: Comparison between Faster R-CNN and YoloV3. <https://arxiv.org/abs/1910.07234>. Accessed 29 Dec 2019
13. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks. *IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI)* **39**(6), 1137–1149 (2017)
14. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: unified, real-time object detection. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 779–788 (2016)
15. Wang, Y., Wang, A., Hu, C.: A novel airplane detection algorithm based on deep CNN. In: Zhou, Q., Gan, Y., Jing, W., Song, X., Wang, Y., Lu, Z. (eds.) *ICPCSEE 2018*. CCIS, vol. 901, pp. 721–728. Springer, Singapore (2018). https://doi.org/10.1007/978-981-13-2203-7_60
16. Wang, Y., Fang, B., Lan, L., Luo, H., Tang, Y.: Adaptive binarization: a new approach to license plate characters segmentation. In: *International Conference on Wavelet Analysis and Pattern Recognition*, pp. 91–99 (2012)
17. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778 (2016)
18. Cortes, C., Vapnik, V.: Support-vector networks. *Mach. Learn.* **20**(3), 273–297 (1995)
19. Evgeniou, T., Pontil, M.: Support vector machines: theory and applications. In: Paliouras, G., Karkaletsis, V., Spyropoulos, C.D. (eds.) *ACAI 1999*. LNCS (LNAI), vol. 2049, pp. 249–257. Springer, Heidelberg (2001). https://doi.org/10.1007/3-540-44673-7_12
20. Rezaeifar, S., Taran, O., Voloshynovskiy, S.: Classification by re-generation: towards classification based on variational inference. In: *26th European Signal Processing Conference (EUSIPCO)*, pp. 2005–2009 (2018)