2018 IEEE 14th International Conference on Control and Automation (ICCA)
June 12-15, 2018. Anchorage, Alaska, USA

Real-Time Landing Spot Detection and Pose Estimation on Thermal Images Using Convolutional Neural Networks

Xudong Chen[1], Feng Lin[1,2], Mohamed Redhwan Abdul Hamid[1], Swee Huat Teo[1], Swee King Phang[3]

***Abstract*— This paper presents a robust, accurate and real-time approach to detect landing spot position and orientation information using deep convolutional neural networks and image processing technique on thermal images. The proposed novel algorithm pipeline consists of two steps: ledge detection and orientation information extraction. The extracted pose information of the landing spot from thermal images could be used to facilitate autonomous operations of unmanned aerial vehicles (UAVs) in both of day and night time. In order to land on the narrow and long ledge, UAV requires accurate orientation information of the ledge. Moreover, the method is scale and rotation invariant and also robust to occlusion in certain special and unexpected situations. Our algorithm runs at 20 frames per second on NVIDIA GTX 1080Ti GPU with the real flight thermal image dataset captured by T-Lion UAV developed by Temasek Laboratories@NUS.**

I. Introduction

In the recent decades, unmanned aerial vehicles (UAVs) are widely utilized in many applications in both military and civilian fields, such as surveillance, exploration, agriculture, search and rescue as well as border patrol. The evolution of autonomous technologies has allowed UAVs to become more intelligent. Automation of landing as one of the most critical phases of UAV flight is still a challenging problem, especially onto an unusual landing spot. To aid the landing process, global positioning system (GPS), LiDAR and radar sensors have been widely used for relative pose estimation between the aircraft and the landing spot.

Due to rich information provided by the images, vision-based autonomous aerial systems are recently becoming popular. Many algorithms and systems [1][2][3] have been proposed and developed to guide the UAV to land on the moving platforms. These systems require fiduciary marker [4] to be placed on the landing sites to indicate the location and orientation. The detection typically relies on the specifically designed algorithms which may only work well with the corresponding markers. Some tracking methods proposed in [5][6] could be utilized to mitigate the computational cost of detection algorithms to increase the frame rate. However, general understanding towards the scene, such as custom object detection, is challenging but important for many vision-based autonomous navigation of robots.

[1]Xudong Chen, Feng Lin, Mohamed Redhwan, Swee Huat Teo are with Temasek Laboratories, National University of Singapore, Singapore. {tslcx, linfeng, tslmor}@nus.edu.sg

[2]Feng Lin are also with Department of Electrical and Computer Engineering, National University of Singapore, Singapore. linfeng@nus.edu.sg

[3]Swee King Phang is with School of Engineering, Taylor's University, 1 Jalan Taylors, 47500, Subang Jaya, Selangor, Malaysia. sweeking.phang@taylors.edu.my

Fig. 1: Experimental flight test using UAV at the real mission site with the onboard thermal camera.

We as a human could easily identify the suitable landing spot from the image; however, robots are not as complex as human beings. In order to execute tasks during both day and night, a thermal camera is employed to provide visual information to localize the UAV. The task is even more challenging due to the limited color information given by thermal images. Qi [7] proposed to use the histogram of sparse code to represent image features and then detect pedestrian using extracted features. The proposed method has good performance but may suffer from occlusion.

Since the introduction of the deep convolutional neural network, object detection is formulated as an object classification problem. The advancement of region proposal methods [8] and region-based convolutional neural networks [9][10] has made the object detection almost real time. Ren and his coworkers [11] proposed Faster R-CNN to use the deep convolutional neural network to compute region proposal to accelerate the inference and proposed regions are detected using Fast R-CNN detector [10]. However, the running frequency of the algorithm on the GPU is varied from 5 to 7 Hz, which is hard for real-time applications that rely on low latency predictions. In order to maximize the detection performance, Redmon [12] uses region proposal network (RPN) in Faster R-CNN to predicts offsets and confidences for anchor boxes, which are easier for the network to learn. They utilized standard k-means on the training set to generate good priors for the anchor box dimensions. The network could achieve 40 frames per second (FPS) on GeForce GTX Titan X with the image resolution of 544×544 pixels.

In this particular mission, the UAV is required to perch on the ledge of the roof in order to respond to and monitor

any emergent situations besides the building, as shown in Fig. 1. The typical roof ledge has a width of 20 cm that could be easily captured by a customized lading gear of the quadcopters. In this paper, we focus on the problem of finding landing ledge given a thermal image view of the scene. Largely inspired by the method proposed by Redmon [12], we designed an algorithm pipeline to localize the ledge from the thermal image and then extract the orientation information using image gradient based line extraction and Hough Transformation.

This paper consists of the following sections: Section II focuses on the ledge detection algorithm. Section III will explain the ledge information extraction using image processing, while Section IV introduces our real flight experiment platform and sensors. The experiment results will be shown in Section V to validate our method. Lastly, future work and conclusion remarks will be made.

II. Ledge Detection using Convolutional Neural Networks

Traditional image analysis usually uses background and foreground segmentation to separate the objects moving in the foreground and to count the properties such as color, shape, position, trajectory as well as quantity of all the objects and then to identify and recognize people or cars with pre-defined thresholds. However, such methods are subject to many uncontrollable noise and external influences, such as lighting, occlusion and haze. Traditional object detection usually consists of two steps: features such as SIFT [13], Haar [14] and HoG [15] are extracted from the input image and then objects are classified based on the feature map [16] using SVM [17] or boosting techniques [18].

Convolutional neural networks (CNNs) have outperformed many traditional methods on computer vision problems in terms of speed and accuracy, especially on some high-level vision issues, such as classification [19], detection [11] and semantic segmentation [20]. With the advance of deep learning, the feature extraction has been rapidly evolved with better deep convolutional architectures. Similarly but in a simple way, the later object classifier uses multi-layer perceptrons to classify objects into various categories. We have chosen YOLOv2 as our detector in the framework as the method is extremely fast compared to other detection algorithms, reaching an average 20 FPS in our application.

A. Classifier

The ledge detection should be fast and accurate enough to recognize the landing spot in real-time. To utilize the huge amount of image data available to classify the object, the ledge detection is modeled as a classification problem. Most state-of-the-art detection methods typically rely on pre-trained classifiers that utilize the classification dataset ImageNet [21]. Based on comparison done by Kim [22], we have compared various state-of-art classifiers and their performance on the classification problem as shown in Table I. For real-time applications, it is always preferred to have shorter inference time for the detection. Without compromising much accuracy, DarkNet-19 is a suitable option for our problem and the structure is shown in Table II.

Type	Filters No.	Size	Output
Conv	32	3×3	224×224
MaxPooling		2×2	112×112
Conv	64	3×3	112×112
MaxPooling		2×2	56×56
Conv	128	3×3	56×56
Conv	64	1×1	56×56
Conv	128	3×3	56×56
MaxPooling		2×2	28×28
Conv	256	3×3	28×28
Conv	128	1×1	28×28
Conv	256	3×3	28×28
MaxPooling		2×2	14×14
Conv	512	3×3	14×14
Conv	256	1×1	14×14
Conv	512	3×3	14×14
Conv	256	1×1	14×14
Conv	512	3×3	14×14
MaxPooling		2×2	7×7
Conv	1024	3×3	7×7
Conv	512	1×1	7×7
Conv	1024	3×3	7×7
Conv	512	1×1	7×7
Conv	1024	3×3	7×7

TABLE II: Model Structure of Darknet-19. The table is based on the paper [12].

Model	Top-1 Accuracy	Top-5 Accuracy	GPU runtime
AlexNet	57.0%	80.3%	1.5 ms
VGG-16	70.5%	90.0%	10.7 ms
GoogleNet	72.5%	90.8%	6.4 ms
ResNet-50	75.8%	92.9%	7.0 ms
DarkNet	61.1%	83.0%	1.5 ms
DarkNet-19	72.9%	91.2%	6.0 ms

TABLE I: Performance of state-of-art classifiers. In this paper, we choose the DarkNet model to achieve real-time detection on our system.

B. Anchor Boxes

It is not optimal to run the classifier on every small patch of the input image. To avoid the high computational cost of sliding window and pyramid methods, a few techniques are proposed, such as selective search by RCNN [9], the grid-cell proposal by YOLO [23] and region proposal network (RPN) by Fast and Faster RCNN [11]. These region proposal methods are better than sliding window but still far behind real-time performance while real-time YOLO lack of certain accuracy.

The YOLOv2 is proposed to use anchor boxes to predict bounding boxes and associate predicts class and objectness for each bounding box. With fine-tuning of the detection structure and pipeline, it has 78.6% mean Average Precision (mAP) while runs at 40 FPS on GeForce GTX Titan X GPU.

C. Training using Thermal Images with Annotation

To use YOLOv2 to detect the landing ledge on the thermal images, we need to train the network to be able to detect such

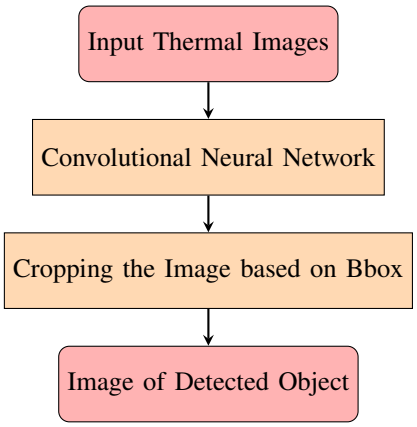


Fig. 2: Flowchart of ledge detection on the thermal images. The detected object image is used to extract ledge information.

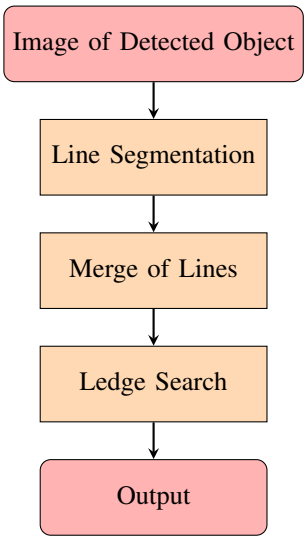


Fig. 3: Flowchart of ledge information extraction.

custom object. In order to acquire decent detection performance on the custom object, we use 149 thermal images with variant scaled and rotated landing ledges. The landing ledges are human annotated to provide bounding box ground truth. The training process begins from pre-trained convolutional weights trained from ImageNet. And with powerful NVIDIA GTX 1080Ti, the loss could easily converge within hours. The training result is shown in Section IV-B.

D. Testing using Onboard Thermal Images

To validate the performance of the retrained networks using our own thermal images and labels, we tested the onboard thermal images. The Fig. 2 shows the flow of the testing process.

III. Ledge Information Extraction using Image Processing

Although the aforementioned ledge detection has already successfully indicated the location of the ledge in the thermal images, its geometrical information for the landing of the UAV, including orientation and width of the ledge, has not been provided yet, which are critical for the safe landing of the UAV. Based on the ledge detection results, we propose an algorithm to efficiently and robustly extract the key geometry of the ledge, including orientation and width, from the detected region in the image.

In fact, the ledge information extraction can be considered as a special case of line segmentation, which has been studied for many years but still challenging in real life applications due to distortion of the lens, non-Lambertian surfaces, dynamic lighting conditions, complex background, etc. In addition, this is the first published work as far as we know that studies the line segmentation using the thermal images. The proposed algorithm consists of three major parts: line segment detection, merge of lines, and ledge search, which has been briefly illustrated in Fig. 3.

A. Line Segmentation

During the phase of ledge detection, we noticed that the ledges vary in appearance based on different lighting conditions and view perspectives. Based on typical ledges on the building's rooftop and a bird's eye point of view, the ledge will appear as near parallel straight lines. In this paper, the ledge considered is limited to such shapes. There are several widely used methods for line segmentation. Traditional methods include Hough transformation [24] which could correct discontinuities in ledge detected to present a clearer image and also formed as a prediction on how far the ledge might extend. Normally, the first step to the line detection will be carried out by the edge detection, such as Canny edge detector, which may cause a lot of false detection, especially in the noisy background. Therefore, we employ the line segment detector (LSD) method proposed in [25] to conduct line segment detection.

LSD is a linear-time line segment detector to give subpixel accurate results without any prior tuning, which could be applied to any form of images. It controls the number of false alarms to an average of only one per image. LSD is aimed at detecting locally straight contours on images and then defined as line segments. In this paper, the LSD algorithm takes the normalized thermal image directly and outputs a list of detected line segments. One example of the line segmentation results is shown in Fig. 4.

B. Merge of Lines

From the above sections, we observe that the detected lines exhibit discontinuities due to sensor noise. We apply the Hough transformation to correct the discontinuities by merging the lines with their neighbors with the same orientation. The lines with the gap less than d_{gap} will be merged together, and those lines with the length less than d_length will be removed to reduce possible false alarms.

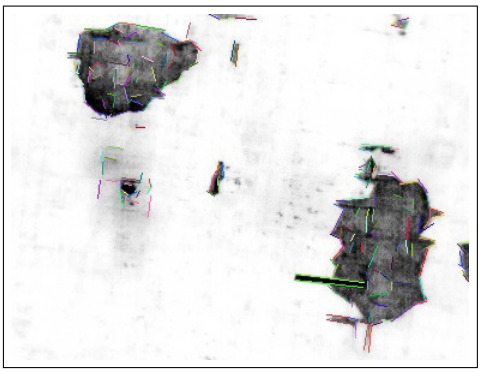


Fig. 4: Illustration of line segmentation using LSD in a noisy environment. The extracted lines have been marked in different colors.

C. Search for the Ledge

We will search for the ledge based on the merged lines. We assume the two edges of the ledge are parallel and the distance between the edges are within a certain range in the image during the automatic landing of the UAV. The brief illustration of searching the ledge is given in Algorithm 1.

Algorithm 1: Search for the ledge algorithm.

Input: A set of detected lines: $\{l_1, l_2, \ldots, l_n\}$
Output: The orientation of the ledge: Ψ_{ledge}

```
for Every two lines do
    Compute their angle difference δ in [0, 90] degree;
    if δ < δ_max then
        Compute their distance d in the image;
        if δ < δ_max and then
        else
            This pair of lines is a candidate for ledge ;
        end
        This pair of lines is not a ledge ;
    else
        This pair of lines is not a ledge ;
    end
end
Choose the longest one from those candidates as the ledge;
Compute the orientation of the ledge in the image: Ψ_ledge ;
```

IV. Experiment

To validate the algorithm pipeline, we have implemented the algorithms on our UAV. Thermal images are collected through real flight trials at an actual building. The algorithms are developed in Robotics Operating System (ROS) for real-time applications.

A. UAV Platform

A customized quadcopter UAV named T-Lion, as shown in Fig. 6, is utilized as the test platform to implement the proposed algorithms. T-Lion UAV is developed by the Control Science Group of Temasek Laboratories at the National University of Singapore. Due to its suitable size and high payload, it has been widely used in many applications.

Fig. 5: Experiment setup in the Vicon room. The UAV is guided to perch on the mock landing ledge to simulate the real life application which is required to land on the rooftop.

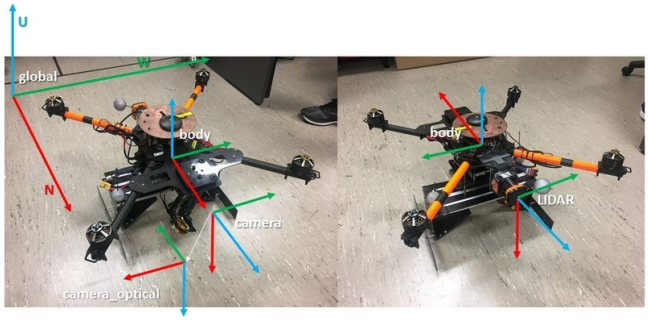


Fig. 6: T-Lion UAV with labelled coordinate systems.

1) Gimbal Controlled Thermal Camera System: To execute the mission during both day and night time, Optris PI-450 thermal camera, as shown in Fig. 7, is employed for visual image capturing. It outputs 382×288 pixels real-time thermographic images at 20 Hz with the temperature range between -20°C up to 900°C. This camera is required to face vertically downwards at all time during the flight. In order to achieve a stabilized visual image capturing without motion blur, an Arris Zhaoyun 2-axis brushless gimbal is adapted to carry and stabilize the camera. The camera is controlled in both pitching and rolling directions. To accommodate the choice of the camera, minor modification is carried out on the mounting plate of the gimbal.

2) Perching Landing Gear: A special landing gear is designed to accommodate the downward facing camera for visual odometry enabled flight with a clamp-like appearance to grab and hold onto common ledges on rooftops in Singapore. The experiment setup in the mocked ledge is shown in Fig. 5 to illustrate the landing process. The landing gear acts as a compression clamp as the UAV lowers itself onto the ledge with precise control utilizing laser scanners. Sufficient clearance from the camera to the roof ledge was considered in the design to prevent damage during the precision landing. The landing gear designed incorporates flat V-Springs made

Fig. 7: Optris Pi-450 thermal camera on the gimbal system. The camera is controlled to face downwards during the flight.

Fig. 8: 3D drawing of T-Lion UAV with the customized landing gear.

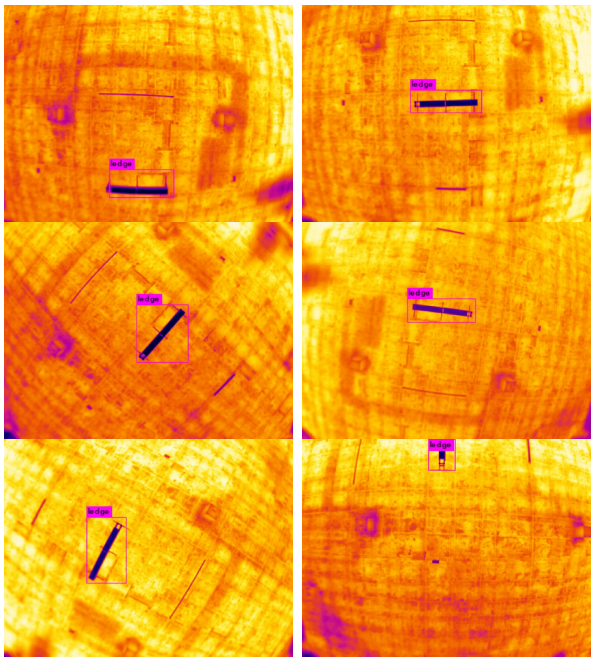


Fig. 9: Ledge detection output on thermal images.

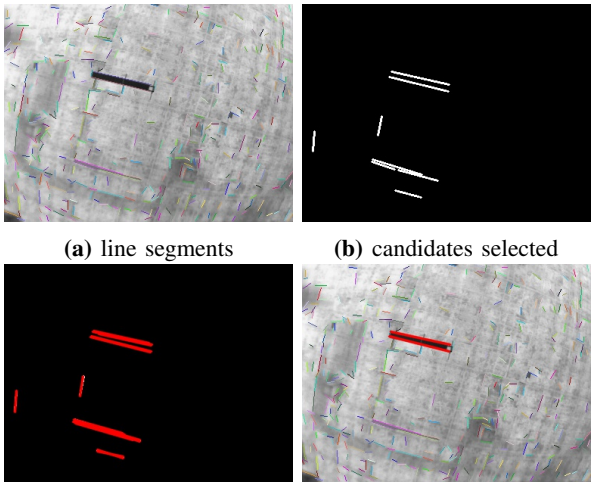
of spring steel angled at 60 degrees. The main part of the landing gear is then made of carbon fiber plates of varying thicknesses with patterned holes to reduce weight which is then attached in a downward C-shape configuration to the UAV as seen in Fig. 8. Longer but narrower carbon fiber plates are attached to the bottom tip of the landing gear to provide better stability on the ground.

B. Ledge Detection Results

To train the model, we utilized pre-trained Darknet-19 weights on ImageNet, and fine-tuned the weights with our own thermal image dataset and labels. With carefully labeled ledge from various perspectives, the convolutional neural network is able to detect ledge with rotation and scale variation and is also robust to occlusion. The detection results are shown in Fig. 9.

The detection network is robust to handle challenging cases, such as rotation, scale and occlusion of the ledge object. In overall, the detection rate is over 95 percent and it could achieve 20 FPS on the NVIDIA GTX 1080 Ti platform.

As shown in Fig. 10, we could easily detect all the line segments from the image using aforementioned methods. Line segments with similar orientation are merged to reduce discontinuity caused by the noise and lighting. The ledge is effectively detected from the candidates and the orientation and position could be easily acquired from the image based



(a) line segments **(b)** candidates selected

(c) merged lines **(d)** final outputs

Fig. 10: The steps of ledge detection based on the line segments and searching algorithm: a) All the line segments detected from thermal image; b) Suitable candidates are selected from line segments based on threshold; c) Short line segments with similar orientations are merged into longer line segments for further step; d) The final ledge detected from the image, orientation and position could be calculated from camera matrix.

on the camera matrix. However, in some cases where there are similar shapes in the image, the simple searching algorithm is not able to reject strong outliers from the image. Hence, the detection algorithm is important and essential to robustly reject the outlier by only searching the line segments inside the bounding box.

V. Conclusions

In conclusion, we present an accurate and fast algorithm pipeline for detecting the landing spot and extracting the pose information in the thermal images. The proposed algorithms combine the state-of-the-art deep learning based object detection and traditional digital image processing. The advantages of two image processing techniques are combined to make the problem easier to be solved. Our pipeline could achieve real-time performance on the modern NVIDIA GTX 1080Ti GPU. In future, we would like to extend the work to the embedded onboard GPU TX1 to facilitate the real-time landing spot detection on the UAV to guide the UAV to land on the rooftop ledge for surveillance.

References

[1] D. Lee, T. Ryan, and H. J. Kim, "Autonomous landing of a vtol uav on a moving platform using image-based visual servoing.," in *IEEE Int. Conf. on Robot. and Autom. (ICRA)*, 2012, pp. 971–976.

[2] X. Chen, S. K. Phang, M. Shan, and B. M. Chen, "System integration of a vision-guided UAV for autonomous landing on moving platform," in *IEEE Int. Conf. on Control & Autom. (ICCA)*, 2016, pp. 761–766.

[3] K. Wang, S. K. Phang, Y. Ke, X. Chen, K. Gong, and B. M. Chen, "Vision-aided tracking of a moving ground vehicle with a hybrid uav," in *IEEE Int. Conf. on Control & Autom. (ICCA)*, 2017, pp. 28–33.

[4] E. Olson, "Apriltag: A robust and flexible multi-purpose fiducial system," University of Michigan APRIL Laboratory, Tech. Rep., May 2010.

[5] M. Lao, Y. Tang, and L. Feng, "Patch-based keypoints consensus voting for robust visual tracking," in *Annual Conf. of the IEEE Ind. Electron. Soc. (IECON)*, 2016, pp. 6109–6115.

[6] M. Lao, Y. Tang, L. Feng, and Y. Li, "Structural keypoints voting for global visual tracking," in *IEEE Int. Conf. on Robot. and Biomimetics (ROBIO)*, 2016, pp. 583–588.

[7] B. Qi, V. John, Z. Liu, and S. Mita, "Pedestrian detection from thermal images: A sparse representation based approach," *Infrared Physics & Technology*, vol. 76, pp. 157–167, 2016.

[8] J. Uijlings, K. van de Sande, T. Gevers, and A. Smeulders, "Selective search for object recognition," *Int. Journal of Comput. Vision*, 2013.

[9] R. B. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," *CoRR*, vol. abs/1311.2524, 2013.

[10] R. Girshick, "Fast r-cnn," in *Proceedings of the 2015 IEEE Int. Conf. on Comput. Vision (ICCV)*, Washington, DC, USA, 2015, pp. 1440–1448.

[11] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in Neural Inform. Process. Syst.*, 2015, pp. 91–99.

[12] J. Redmon and A. Farhadi, "YOLO9000: better, faster, stronger," *CoRR*, vol. abs/1612.08242, 2016.

[13] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vision*, vol. 60, no. 2, pp. 91–110, Nov. 2004.

[14] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," *Int. J. of Comput. Vision*, pp. 511–518, 2001.

[15] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *IEEE Conf. on Comput. Vision and Pattern Recognition (CVPR)*, vol. 1, 2005, pp. 886–893.

[16] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 9, pp. 1627–1645, Sep. 2010.

[17] C. J. C. Burges, "A tutorial on support vector machines for pattern recognition," *Data Min. Knowl. Discov.*, vol. 2, no. 2, pp. 121–167, Jun. 1998.

[18] R. E. Schapire, "A brief introduction to boosting," in *Proceedings of the 16th Int. Joint Conf. on Artificial Intell. - Volume 2*, Stockholm, Sweden, 1999, pp. 1401–1406.

[19] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Inform. Process. Syst.* 2012, pp. 1097–1105.

[20] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *IEEE Conf. on Comput. Vision and Pattern Recognition (CVPR)*, 2015, pp. 640–651.

[21] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A Large-Scale Hierarchical Image Database," in *IEEE Conf. on Comput. Vision and Pattern Recognition (CVPR)*, 2009.

[22] Y. Kim, E. Park, S. Yoo, T. Choi, L. Yang, and D. Shin, "Compression of deep convolutional neural networks for fast and low power mobile applications," *CoRR*, vol. abs/1511.06530, 2015.

[23] J. Redmon, S. K. Divvala, R. B. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," *CoRR*, vol. abs/1506.02640, 2015.

[24] R. O. Duda and P. E. Hart, "Use of the hough transformation to detect lines and curves in pictures," *Commun. ACM*, vol. 15, no. 1, pp. 11–15, Jan. 1972.

[25] R. G. von Gioi, J. Jakubowicz, J.-M. Morel, and G. Randall, "Lsd: A fast line segment detector with a false detection control.," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 4, pp. 722–732, 2010.