Object Detection Using Convolutional Neural Networks

Reagan L. Galvez
Gokongwei College of Engineering
De La Salle University
Manila, Philippines
reagan_galvez@dlsu.edu.ph

Argel A. Bandala
Gokongwei College of Engineering
De La Salle University
Manila, Philippines
argel.bandala@dlsu.edu.ph

Elmer P. Dadios
Gokongwei College of Engineering
De La Salle University
Manila, Philippines
elmer.dadios@dlsu.edu.ph

Ryan Rhay P. Vicerra
Gokongwei College of Engineering
De La Salle University
Manila, Philippines
ryan.vicerra@dlsu.edu.ph

Jose Martin Z. Maningo
Gokongwei College of Engineering
De La Salle University
Manila, Philippines
jose.martin.maningo@dlsu.edu.ph

***Abstract*—Vision systems are essential in building a mobile robot that will complete a certain task like navigation, surveillance, and explosive ordnance disposal (EOD). This will make the robot controller or the operator aware what is in the environment and perform the next tasks. With the recent advancement in deep neural networks in image processing, classifying and detecting the object accurately is now possible. In this paper, Convolutional Neural Networks (CNN) is used to detect objects in the environment. Two state of the art models are compared for object detection, Single Shot Multi-Box Detector (SSD) with MobileNetV1 and a Faster Region-based Convolutional Neural Network (Faster-RCNN) with InceptionV2. Result shows that one model is ideal for real-time application because of speed and the other can be used for more accurate object detection.**

Keywords— computer vision, convolutional neural networks, image classification, object detection, transfer learning

I. INTRODUCTION

Object detection is important in computer vision systems. It can be used for many applications like video surveillance [1], medical imaging [2], and robot navigation [3]. Many algorithms can be used for this task like background subtraction, temporal differencing, optical flow, Kalman filtering, support vector machine, and contour matching [4]. Aside from the said algorithms, the newest method used for object detection is called convolutional neural networks (CNN). Breakthroughs in image classification started when Alex [5] won the 2012 ImageNet competition using deep convolutional neural networks. They trained a deep CNN to classify 1.2 million high resolution images in the ImageNet [6] contest that has 1000 categories. They achieved more accurate prediction than the previous state of the art models. From this, many researchers became interested in finding a novel way to develop an efficient deep convolutional neural networks.

There are some recent approaches for object detection, in paper [7], the authors developed a low shot transfer detector using a flexible deep architecture and a regularized transfer learning framework to address object detection using few training data. Another paper in [8] proposed a region selection network and a gatting network for object detection. The region selection network serves as guidance on where to select regions to learn the features from. On the other hand, gatting network serves as local feature selector that transforms feature maps. In [9] used convolutional neural networks for visual target tracking. They created an ad-hoc large dataset with positive and negative examples of framed objects from ImageNet database and selected the most promising patch using AlexNet [5] architecture. Another method used for object detection is active learning [10]. This is a class of algorithms that searches for the most informative samples to include in a training dataset that is effective in image classification. Lastly, in [11] used artificial neural networks to detect objects by shape and color pattern recognition.

This paper is organized as follows; Section II is a brief introduction about convolutional neural networks. Section III is discusses the concept of transfer learning. Section IV discusses the different TensorFlow models used for object detection. Section V shows the experiment set-up as well as the information about the dataset. Section VI is the results discussion.

II. CONVOLUTIONAL NEURAL NETWORKS

Convolutional neural network (CNN) is a class of deep, feed-forward artificial neural network that has been utilized to produce an accurate performance in computer vision tasks, such as image classification and detection [5]. CNNs are like traditional neural network, but with deeper layers. It has weights, biases and outputs through a nonlinear activation. The neurons of the CNN are arranged in a volumetric fashion such as, height, width and depth.

Fig. 1 shows the CNN architecture, it is composed of convolutional layer, pooling layer and fully connected layer. Convolutional layer and pooling layer are typically alternated and the depth of each filter increases from left to right while the output size (height and width) are decreasing. The fully connected layer is the last stage which is similar to the last layer of the conventional neural networks.

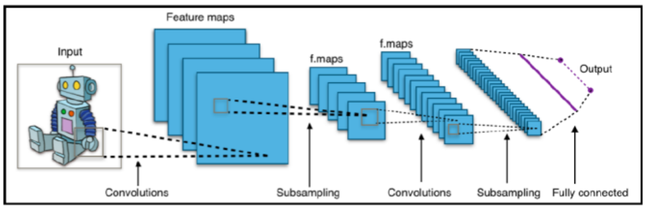

Fig. 1. CNN Architecture [12]

Authorized licensed use limited to: M S RAMAIAH INSTITUTE OF TECHNOLOGY. Downloaded on May 21,2024 at 05:31:17 UTC from IEEE Xplore. Restrictions apply.

978-1-5386-5457-6/18/$31.00 ©2018 IEEE

The input is an image that will hold pixel values. It has three dimensions such as width, height and depth (RGB channels) example is [50 x 50 x 3] [13]. The convolutional layer will compute the output of neurons that are connected to local regions in the input. The layer's parameters are composed of a set of learnable filters (or kernels), which convolved across the width and height of the input volume extending through its depth, computing the dot product between the entries of the input and the filter. This produces a 2-dimensional activation map of that filter and as a result, the network learns filters that trigger when it detects some particular type of feature at some spatial position in the input. The function called Rectified Linear Unit (ReLU) layer will perform elementwise activation function. ReLU is defined in (1),

$$f(x) = \max(0, x) \qquad (1)$$

This function is zero for negative values and grows linearly for positive values. This will not affect the volume size. The pooling layer outputs the maximum activation in a region. This down samples the spatial dimensions such as width and height. The output layer is the fully connected layer which is similar to the final layer of the neural network. This layer used commonly used softmax activation to output probability distributions over the number of output classes.

III. Transfer Learning

Transfer learning is a powerful deep learning technique in which pre-trained models can be used for feature extraction and fine tuning. This technique can be used in image classification like vehicle classification [14], object detection and segmentation. The advantage of using this technique is it saves time to train the network from the start and less data is needed to get good results. The training can also be done using a central processing unit (CPU) even without the computational power of graphics processing unit (GPU). Instead of creating new model from scratch, pre-trained models can be used. These models are trained from the large image database like ImageNet [6] and COCO [15] dataset. Some of these models are AlexNet [5], VGG16, VGG19 [16], ResNet50 [17], InceptionV2, InceptionV3 [18], Xception [19], DenseNet [20] and MobileNet [21]. This paper focuses on the two models only, InceptionV2 and MobileNet.

IV. Tensorflow Object Detection Models

TensorFlow is an open-source software library for high performance numerical computation [22]. The following TensorFlow models are used to detect quadrotor unmanned aerial vehicle (UAV) [23] and person in an image and video.

A. SSD with MobileNetV1

A Single Shot Multi-Box Detector (SSD) is an object detection approach in images using a deep neural network [24]. It produces bounding box and class scores of the detected object at greater speed than previous approach like You Only Look Once (YOLO) [25]. On the other hand, MobileNet is a deep learning model that can be applied to various recognition tasks like object detection, finegrain classification, landmark recognition, and face attributes [21]. The advantages of MobileNet are accurate, fast, small and easy to tune. It is based on a simplified architecture that uses depthwise separable convolutions to build light weight deep neural networks [21]. SSD models that used MobileNet in object detection are lightweight so that in can be deployed in mobile devices and can be run in real-time [26].

B. Faster-RCNN with InceptionV2

Faster-RCNN is a single, unified network for object detection. It uses region proposal network (RPN) module that serves as the attention mechanism [27]. It tells the unified network where to look. On the other hand, Inception is composed of efficient inception module with 22 layers and no fully connected layers [18]. The main advantage of this model is the improved utilization of the computing resources inside the network. Inception module is a network within a network and modules are stack on top of each other. It has 5 million parameters that are 12 times less than AlexNet model. Faster-RCNN together with InceptionV2 is computationally intensive but produces more accurate results in object detection.

V. Experiment Setup

To implement the object detection using CNN, TensorFlow Object detection API [28] was used. This is an open source framework for constructing, training and deploying object detection models. The dataset used for this research was limited to a person and quadrotor only. Training images were extracted from personal video while testing images were from the internet. Table I shows the number of images used for training and testing. As a rule of thumb, 80% of the images were utilized for training and 20% for testing. The images were labeled manually using LabelImg, a graphical image annotation tool [29]. This creates XML files in PASCAL VOC format [30].

TABLE I. Number of Images Per Category

Categories	Number of Images		
	Training	Testing	Total
Person and Quadrotor	355	89	444

Table II shows the number of labels per category. Number of labels was different in every image depending of the extracted frame.

TABLE II. Number of Labels Per Category

Categories	Number of Labels		
	Training	Testing	Total
Person	461	67	528
Quadrotor	679	95	774

Authorized licensed use limited to: M S RAMAIAH INSTITUTE OF TECHNOLOGY. Downloaded on May 21,2024 at 05:31:17 UTC from IEEE Xplore. Restrictions apply.

Fig. 2 shows the sample training image with four quadrotors. Bounding box was drawn manually using LabelImg software.

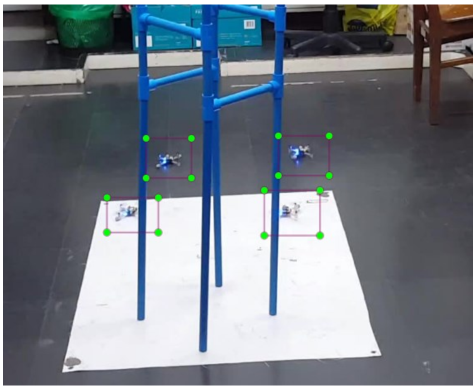

Fig. 2. Sample labeled image

VI. Results and Discussion

The first TensorFlow model used for training was SSD with MobileNetV1. The total number of steps in the training was 30,113 steps. Fig. 3 shows the total loss versus number of steps of the training using TensorBoard, a visualization tool for machine learning. The maximum and minimum loss was 17.63 and 1.17 respectively. Minimum loss is desired and a decreasing value means the model is learning during the training. Training can be stopped anytime if the loss is not decreasing anymore. During the training, it periodically saves a checkpoint every five minutes. This checkpoint will be used to export inference graph (.pb file) that contains graph variables frozen as constants.

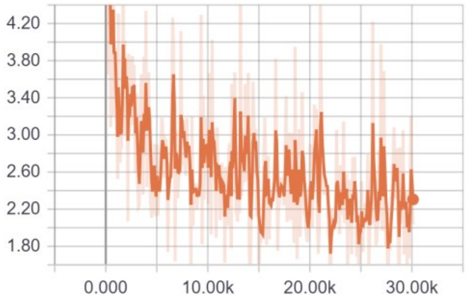

Fig. 3. Total loss versus number of steps using SSD with MobileNetV1

Fig. 4 shows the total loss versus time. Training the model took 11 hours using NVIDIA GeForce GTX 1050. This will vary depending on the GPU power of the computer.

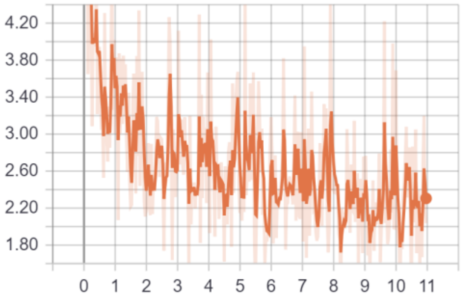

Fig. 4. Total loss versus time (hours) using SSD with MobileNetV1

After the training, inference graph was exported. Inference graph contains the object detection classifier. Fig. 5 shows the sample detection in an image with 5 persons and 1 quadrotor. The SSD with MobileNetV1 failed to detected 5 persons and 1 quadrotor. Although this model has large loss, its high speed detection can be applied in real-time application.

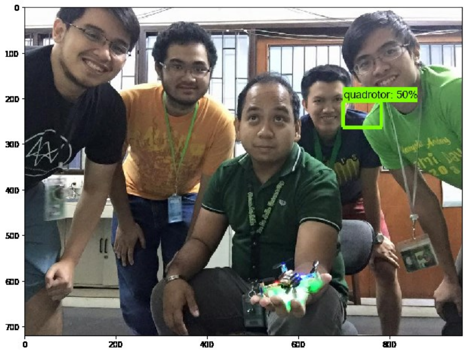

Fig. 5. Detection testing with 5 persons and 1 quadrotor using SSD with MobileNetV1

Fig. 6 shows another test image to measure the model's performance. The model failed to detect the 2 quadrotors.

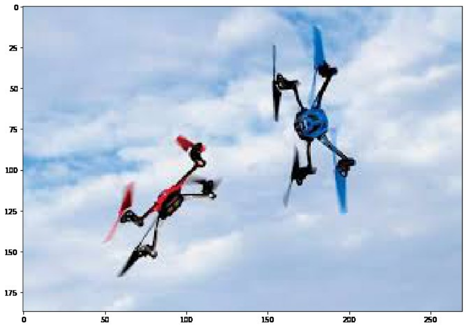

Fig. 6. Detection testing with 2 quadrotors using SSD with MobileNetV1 [31]

The next TensorFlow model used for training was Faster-RCNN with InceptionV2. The total number of steps in the training was 68,122 steps. This model has a lower loss than the SSD with MobileNetV1. The maximum and minimum loss was 2.39 and 0.01 respectively. Fig. 7 shows the total loss versus number of steps of the whole training.

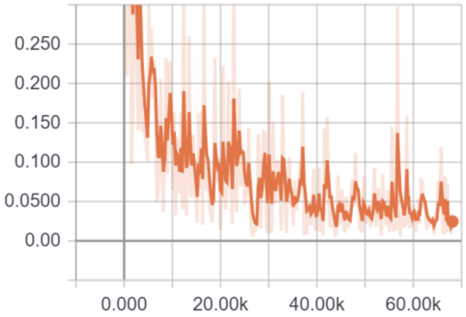

Fig. 7. Total loss versus number of steps using Faster RCNN with InceptionV2

Fig. 8 shows the total loss versus time. Training using Faster-RCNN model took 8 hours using GPU.

Authorized licensed use limited to: M S RAMAIAH INSTITUTE OF TECHNOLOGY. Downloaded on May 21,2024 at 05:31:17 UTC from IEEE Xplore. Restrictions apply.

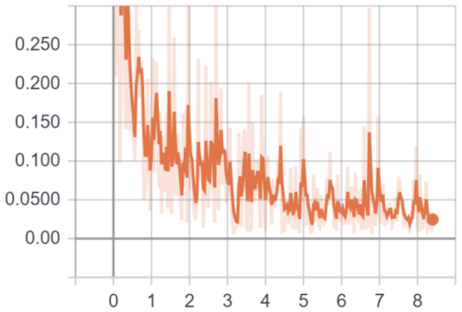

Fig. 8. Total loss versus time (hours) using Faster RCNN with InceptionV2

Fig. 9 shows the sample test image using Faster-RCNN with InceptionV2. The model successfully detected 5 persons and 1 quadrotor in the image but has some false positives. This is still a good detection even the training data is small (355 images).

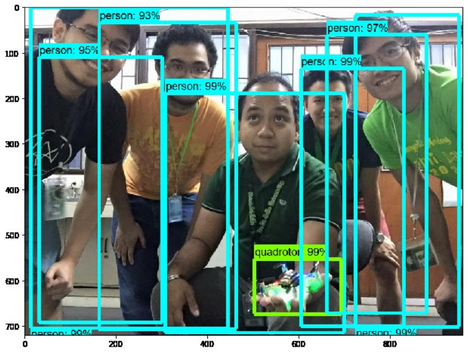

Fig. 9. Detection testing with 5 persons and 1 quadrotor using Faster-RCNN with InceptionV2

In Fig. 10, shows the test image with 2 quadrotors. The model successfully detected all quadrotors.

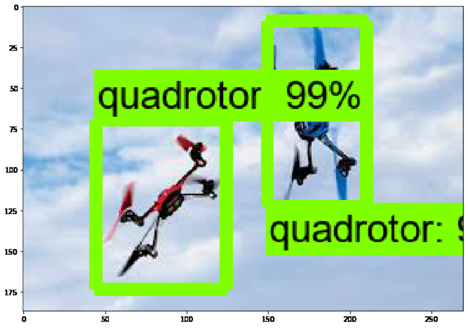

Fig. 10. Detection testing with 2 quadrotors using Faster-RCNN with InceptionV2 [31]

Table III shows the loss comparison of two TensorFlow models during the training. Faster-RCNN with InceptionV2 has lower average loss than SSD with MobileNetV1.

TABLE III. LOSS COMPARISONS

Model	Minimum loss	Maximum loss	Average loss
SSD with MobileNetV1	1.17	17.63	2.72
Faster-RCNN with InceptionV2	0.01	2.39	0.09

Table IV shows the mean average precision (mAP) of the two models using the test set (89 images). The mAP's was based from detection evaluation metrics used by COCO [15]. The mAP @ IOU = 0.50:0.95 used 10 IOU thresholds with 0.05 interval. This IOU averaging is primary challenge metric used by COCO that rewards detectors with better localization. Other IOU thresholds were also used such as 0.50 (PASCAL VOC metric) and 0.75 (strict metric).

TABLE IV. MEAN AVERAGE PRECISION COMPARISONS

Model	mAP IOU = 0.50:0.95	mAP IOU = 0.50	mAP IOU = 0.75
SSD with MobileNetV1	0.124	0.329	0.052
Faster-RCNN with InceptionV2	0.455	0.817	0.440

Table V shows the model size comparisons of the two models. The model was tested in a sample video and compared its speed of detection. The SSD with MobileNetV1 has greater detection speed than Faster-RCNN with InceptionV2 when loaded to a video. But in terms of accurate detection, Faster-RCNN with InceptionV2 is more accurate.

TABLE V. MODEL SIZE COMPARISONS

Model	Size (KB)
SSD with MobileNetV1	22,140
Faster-RCNN with InceptionV2	50,963

VII. CONCLUSION

The object detection capability of the two state of the art models in CNN was successfully demonstrated. It shows that the SSD with MobileNetV1 has high speed detection but low accuracy compared with Faster-RCNN with InceptionV2 that has low speed but more accurate. Base on the results of the experiments, there's a trade-off between accuracy and speed. If we want fast detection capability especially in real-time applications, use SSD with MobileNetV1. If high accurate detection capability is desired, use Faster-RCNN with InceptionV2. For future research, the two models will be implemented as vision system of bomb disposal robot to detect improvised explosive devices (IED's).

ACKNOWLEDGMENT

The author would like to thank Department of Science and Technology – Engineering Research and Development for Technology (DOST-ERDT) and De La Salle University for the financial support while doing the study.

Authorized licensed use limited to: M S RAMAIAH INSTITUTE OF TECHNOLOGY. Downloaded on May 21,2024 at 05:31:17 UTC from IEEE Xplore. Restrictions apply.

REFERENCES

[1] V. Gajjar, A. Gurnani and Y. Khandhediya, "Human Detection and Tracking for Video Surveillance: A Cognitive Science Approach," in 2017 IEEE International Conference on Computer Vision Workshops, 2017.

[2] M. Adel, A. Moussaoui, M. Rasigni, S. Bourennane and L. Hamami, "Statistical-Based Tracking Technique for Linear Structures Detection: Application to Vessel Segmentation in Medical Images," IEEE Signal Processing Letters, vol. 17, no. 6, pp. 555-558, June 2010.

[3] X.-T. Truong, V. N. Yoong and T.-D. Ngo, "RGB-D and Laser Data Fusion-based Human Detection and Tracking for Socially Aware Robot Navigation Framework," in IEEE Conference on Robotics and Biomimetics, Zhuhai, China, 2015.

[4] H. S. Parekh, D. G. Thakore and U. K. Jaliya, "A Survey on Object Detection and Tracking Methods," International Journal of Innovative Research in Computer and Communication Engineering, vol. 2, no. 2, pp. 2970-2978, February 2014.

[5] A. Krizhevsky, I. Sutskever and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in Advances in neural information processing systems, 2012.

[6] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li and L. Fei-Fei, "ImageNet: A Large-Scale Hierarchical Image Database," in 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 2009.

[7] H. Chen, Y. Wang, G. Wang and Y. Qiao, "LSTD: A Low-Shot Transfer Detector for Object Detection," arXiv:1803.01529v1, 2018.

[8] H. Xu, X. Lv, X. Wang, Z. Ren and R. Chellappa, "Deep Regionlets for Object Detection," arXiv:1712.02408v1, December 2017.

[9] "Deep learning to frame objects for visual target tracking," Engineering Applications of Artificial Intelligence, vol. 65, pp. 406-420, October 2017.

[10] C.-C. Kao, P. Sen, T.-Y. Lee and M.-Y. Liu, "Localization-Aware Active Learning for Object Detection," arXiv:1801.05124v1, January 2018.

[11] J. P. N. Cruz, M. L. Dimaala, L. G. L. Francisco, E. J. S. Franco, A. A. Bandala and E. P. Dadios, "Object Recognition and Detection by Shape and Color Pattern Recognition Utilizing Artificial Neural Networks," in 2013 International Conference of Information and Communication Technology (ICoICT), Bandung, Indonesia, 2013.

[12] A. Gulli and S. Pal, Deep Learning with Keras, Birmingham: Packt, 2017.

[13] A. Karpathy, October 2017. [Online]. Available: http://cs231n.github.io/convolutional-networks/.

[14] R. L. Galvez, N. N. F. Giron, M. K. Cabatuan and E. P. Dadios, "Vehicle Classification Using Transfer Learning in Convolutional Neural Networks," in 2017 2nd Advanced Research in Electrical and Electronic Engineering Technology (ARIEET), Jakarta, Indonesia, 2017.

[15] T.-Y. Lin, J. Hays, M. Maire, P. Perona, S. Belongie, D. Ramanan, L. Bourdev, L. Zitnick, R. Girshick and P. Dollar, "Microsoft COCO: Common Objects in Context," arXiv:1405.0312v3, pp. 1-15, February 2015.

[16] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks For Large-Scale Image Recognition," in ICLR, 2015.

[17] K. He, X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition," 2015.

[18] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke and A. Rabinovich, "Going Deeper with Convolutions," arXiv:1409.4842v1, pp. 1-12, September 2014.

[19] F. Chollet, "Xception: Deep Learning with Depthwise Separable Convolutions," 2017.

[20] G. Huang, Z. Liu, L. Van der Maaten and K. Q. Weinberger, "Densely Connected Convolutional Networks," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017.

[21] A. G. Howard, M. Zhu, B. Chen and D. Kalenichenko, "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision," 2017.

[22] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, M. Kudlur, J. Levenberg, R. Monga, S. Moore, D. G. Murray, B. Steiner, P. Tucker, V. Vasudevan, P. Warden, M. Wicke, Y. Yu and X. Zheng, "TensorFlow: A system for large-scale machine learning," arXiv:1605.08695v2, pp. 1-18, May 2016.

[23] R. L. Galvez, G. E. U. Faelden, J. M. Z. Maningo, R. C. S. Nakano, E. P. Dadios, A. A. Bandala, R. R. P. Vicerra and A. H. Fernando, "Obstacle Avoidance Algorithm for Swarm of Quadrotor Unmanned Aerial Vehicle Using Artificial Quadrotor Unmanned Aerial Vehicle Using Artificial," in 2017 IEEE Region 10 Conference (TENCON), Penang, Malaysia, 2017.

[24] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu and A. C. Berg, "SSD: Single Shot MultiBox Detector," arXiv:1512.02325v5, pp. 1-17, December 2016.

[25] J. Redmon, S. Divvala, R. Girshick and A. Farhadi, "You Only Look Once:Unified, Real-Time Object Detection," arXiv:1506.02640v5, pp. 1-10, May 2016.

[26] J. Huang and V. Rathod, June 2017. [Online]. Available: https://research.googleblog.com/2017/06/supercharge-your-computer-vision-models.html.

[27] S. Ren, K. He, R. Girshick and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," arXiv:1506.01497v3, pp. 1-14, January 2016.

[28] J. Huang, A. Fathi, V. Rathod, I. Fischer, C. Sun, Z. Wojna, M. Zhu, Y. Song, A. Korattikara, S. Guadarrama and K. Murphy, "Speed/accuracy trade-offs for modern convolutional object detectors," arXiv:1611.10012v3, pp. 1-21, April 2017.

[29] T. Darrenl, January 2018. [Online]. Available: https://github.com/tzutalin/labelImg.

[30] M. Everingham, L. V. Gool, C. K. I. Williams, J. Winn and A. Zisserman, "The PASCAL Visual Object Classes (VOC) Challenge," International Journal of Computer Vision, vol. 88, no. 2, pp. 303-338, 2010.

[31] J. Brown, 2018. [Online]. Available: http://mydronelab.com/best-pick/quadrotor-drone.html. [Accessed 5 September 2018].

Authorized licensed use limited to: M S RAMAIAH INSTITUTE OF TECHNOLOGY. Downloaded on May 21,2024 at 05:31:17 UTC from IEEE Xplore. Restrictions apply.