Flame Detection Using Deep Learning

Dongqing Shen, Xin Chen, Minh Nguyen, Wei Qi Yan
Auckland University of Technology, Auckland 1010 New Zealand
e-mail: 466801082@qq.com

Abstract—**Flame detection is an increasingly important issue in intelligent surveillance. In fire flame detection, we need to extract visual features from video frames for training and test. Based on them, a group of shallow learning models have been developed to detect flames, such as color-based model, fuzzy-based model, motion and shape-based model, etc. Deep learning is a novel method which could be much efficient and accurate in flame detection. In this paper, we use YOLO model to implement flame detection and compare it with those shallow learning methods so as to determine the most efficient one for flame detection. Our contribution of this paper is to make use of the optimized YOLO model for flame detection from video frames. We collected the dataset and trained them using Google platform TensorFlow, the obtained accuracy of our proposed flame detection is up to 76%.**

Keywords-flame detection; image processing; deep learning

I. INTRODUCTION

Fire detection through digital video and image processing could save manual work by using machine intelligence. For decades, the fire detection was associated with smoking alarms which inspected fire by counting the density of little particles and smoke dust. The method was proved to have low accuracy accompanied by too many false alarms [1] and wasted an enormous amount of social resources. It may miss the real fire incidents, or the detection is too late for an efficient fire extinction. It is also not suitable for fire detection happening in a large area such as forests, farms, buildings or oil tanks.

Using Closed Circuit Television (CCTV) for fire detection could lessen security staff's human labor; hence, it is appropriate for fire detection in a massive area. The networked sensors could capture fire flames [2] and activate the mechanism of incident response.

Visual feature model is a set of methods that allow a camera to fetch visual features of a fire flame such as color-based model, motion- based model [3], shape-based model [4,5], fuzzy-based model [6], hidden Markov model [7], and so on.

Color-based model is to extract fire colors and identify flames from digital images [8,9]. Motions and shapes are two essential features of a fire flame. The Fuzzy pattern is proposed for fire detection. In the fuzzy, motion region of a flame will be set as the Region of Interest (ROI). Fuzzy patterns will judge a fire flame. In flame detection, the detection time is also very crucial [10]. We believe a single feature cannot improve the accuracy of fire detection too much; however, a combination of various features together could increase the precision significantly [11]. Furthermore, the algorithms of deep learning is much fitful for a high precision flame detection.

Deep learning is one of the latest concepts based on artificial neural networks [12]. With the assistance of training set of digital images, the learning process will be encapsulated in a black box to identify fire flames which could increase the accuracy of fire detection. In deep learning, the detection process does not rely on visual features and classifiers like shallow learning; the model is end to end. Convolutional Neural Networks (ConvNets or CNN) is one of the most useful neural networks for object detection [12].

This paper aims at finding out an efficient algorithm for flame detection which will be implemented by using deep neural networks. For deep learning, we will use YOLO model to see how deep learning methods could be applied to fire detection. You Only Look Once (YOLO) is a state-of-the-art and real-time model for object detection. The original YOLO model has 24 layers. Therefore, we will optimize it because we assume that we only have one object to detect simultaneously instead of many. Finally, we are going to compare the performance of the methods for flame detection. We will use the optimized YOLO model to verify how deep learning works in flame detection.

Our related work will be introduced in Section II; the methodology will be explained in Section III, experimental results and analysis will be depicted in Section IV, conclusion and future work will be envisioned in Section V.

II. LITERATURE REVIEW

Besides color model, there are many visual features for flame detection, such as motion-based model, fuzzy-based model, shape-based model, etc. However, most previous studies did not only use one model for flame detection. Multiple feature models have been applied to increase the accuracy of fire detection. One method not only uses color model or motion model but also analyzes temporal variations of flames intensity. The color-based model [13] could have a very high performance. The novel method used color features and back-propagation neural network to classify the features of smoking; motion features by using optical flow are adopted in this method. Support vector machine (SVM) could classify fire pixels in an image, which makes the output of this classification more robust against noise.

Deep learning allows computational models to learn the representations of visual data with multiple layers of abstraction. Deep learning is a new branch of machine learning based on neural networks. Comparing to the BP

978-1-5386-6338-7/18/$31.00 ©2018 IEEE

algorithm [14], a network in deep learning was proposed with deeper layers.

Deep learning has become a very efficient way to resolve visual problems [15] which has been used in multiple areas such as object detection, object recognition and tracking [16]. As same as other objects, deep learning could be used to detect flames which is more efficient compared to those shallow learning models.

Deep learning adopts different training mechanism to deal with the problems existing in traditional neural networks. With initialized weights, deep learning algorithms calculate the output of current networks and optimize the parameters from the former layers; whereas, using back propagation may cause the issue of gradient diffusion in deep neural networks. On the contrary, deep learning adopts one layer-wise mechanism to resolve this problem.

A. *Convolutional Neural Networks*

ConvNets have become one efficient method for image classification [17,18,19], speech recognition [20] and graphics recognition [21]. The deep learning network with shared weights is more like a biological network which decreases the complexity of network model and the amount of weights [22, 23].

In Fig.1, fourteen patterns were proposed for the CNN. With added weights, summating every four pixels in images, we obtained feature maps through an activation function (Sigmoid function, Tanh function, or ReLU). In this paper, the Leaky ReLU will be used as the activation function [24]. In Fig. 1, all the feature maps will be filtered again to obtain C_2 and S_2. At last, all the pixels will be rasterized to the traditional neural networks so as to acquire the result [25].

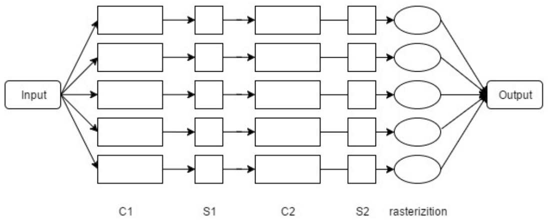


Figure 1. Basic model for CNN.

In CNN, all these neural units will share the same parameters (weight vector and bias) to form a feature map, which means that all neurons are adopted using the same feature in one convolutional layer. With regard to one 1000×1000 image with the filter set as 10×10, one neuron will have 100 parameters and these parameters are all the same; then, no matter how many hidden layers we have, there will be only 100 parameters for the connectivity between two layers. However, this kind of features could decrease the precision. To resolve this problem, the CNN set multiple filters. If there are 100 filters for the 1000×1000 image, there will be 10^4 parameters.

In practice, 3×3 and 5×5 are mostly used as the size of the convolutional kernel. In practical operations, we may add paddings to maintain the image size. For example, we add one padding for 3×3 kernel and two paddings for the 5×5 kernel.

The pooling layer of CNN is for compressing features and extracting the main features which could work for downsampling to simplify the complexity of this network. There are many pooling methods for the CNN, such as fractional max pooling, overlapping pooling and spatial pyramid pooling [26, 27].

B. *YOLO*

YOLO is one of the fast object detectors. Unlike other methods, YOLO creates an S×S grid cells, and each cell will be responsible for the object which falls into the cell. Every cell will predict the bounding box and the confidence score of this box. For evaluating YOLO model, a 7×7 bounding box and 20 labelled classes are defined, which means that it only extracts 98 proposals.

YOLO used the whole image instead of a regional proposal to train and test. When compared object detection to a real-time model, YOLO has an overwhelming advantage. YOLO could reach 45 *fps*, which is much faster than others [28,29,30,31,32].

III. Our Methodology

Flame detection has a higher speed which makes YOLO as the best one. Considered that a flame is not such a small object in the whole image, which makes YOLO may have a better performance of accuracy, YOLO may be one of the best choices for flame detection.

The original YOLO uses 24 convolution layers with two fully connected layers. There is one fast version of YOLO (Fast YOLO) which has fewer layers with the same training and test parameters as YOLO. In flame detection, there is only one class needed to be detected. Hence, the classification would be a fire flame or not, instead of several different objects.

Meanwhile, for the single flame detection, we expect to simplify YOLO network. We use nine convolution layers and one more layer for pre-training; without a fully connected layer, we directly add the network for finding out the four parameters. Each convolution layer will be followed by the max pooling layer and one activation function ReLU.

The original YOLO network has 24 convolutional layers, which is robust to detect multiple objects. In this paper, there is only one object to be detected, which means that we do not need the same complex model as the original YOLO model. In our model, we design a network having 12 convolutional layers.

First of all, we design nine pre-training convolutional networks for locating which grid has a fire flame. The *conv6* has four convolutional layers. After pre-training, the formal training will be carried out. The 7×7×1024 feature vector from *con6* will be calculated after two more convolutional layers are set to a 7×7×516 feature vector. Formal training is used to increase the prediction accuracy of the bounding box.

For other layers in this model, "predict" is used for estimating the bounding box, "pre-valid" is applied to calculate the accuracy of pre-trained and the "valid" is for calculating the precision of formal training.

IOU is the coincidence rate between the predicted face location and the original face region, and the calculation is the intersection of the detected region (DR) and the ground truth (GT), as shown in Fig. 2.

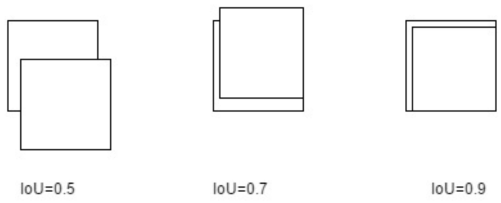


Figure 2. Using IoU to define the accuracy of bounding box.

In training phase, we have two different training methods: pre-training and formal training. YOLO model segmented an image into 7×7 regions. The pre-training is a kind of classification, which is used to find out which grid is its center of the detected object.

For training dataset, we prepare 1720 images of fire flames. We set every 172 images as one unit and trained 20 epochs for the pre-training phase and 40 epochs for the formal training phase.

The first nine convolutional networks output one 7×7×1024 feature vector. Through one more convolution in "conv$_{pre}$", it will become a 49 dimensions vector. The possibilities of positioning in the center grid will be set as $\mathbf{y}=(y_1,y_2,y_3,\ldots,y_{49})$. After normalized them by using the Softmax, we get $\mathbf{p}=(p_1,p_2,p_3,\ldots, p_{49})$. The p_i is presented as eq.(1).

$$p_i = \frac{\exp(y_i)}{\sum_{k=1}^{49} \exp(y_k)} \tag{1}$$

If the ground truth is in the j-th grid, then we will get q_j=1 from $\mathbf{q}=(q_1,q_2,q_3\cdots,q_{49})$. The loss function of pre-training is shown as eq. (2).

$$L_{pre} = \sum_{k=1}^{49} -q_k \log p_k \tag{2}$$

The pre-training aims at minimizing loss function L_{pre} and increasing the accuracy of prediction. The second phase is formal training. Formal training is to ensure the width and height of the bounding boxes. Let us assume that the width and height of an image are (w, h) and (w', h') for the ground truth. The loss function is show as eq. (3).

$$L = (\mathrm{w}- w')^2 + (h-h')^2+ L_{pre} \tag{3}$$

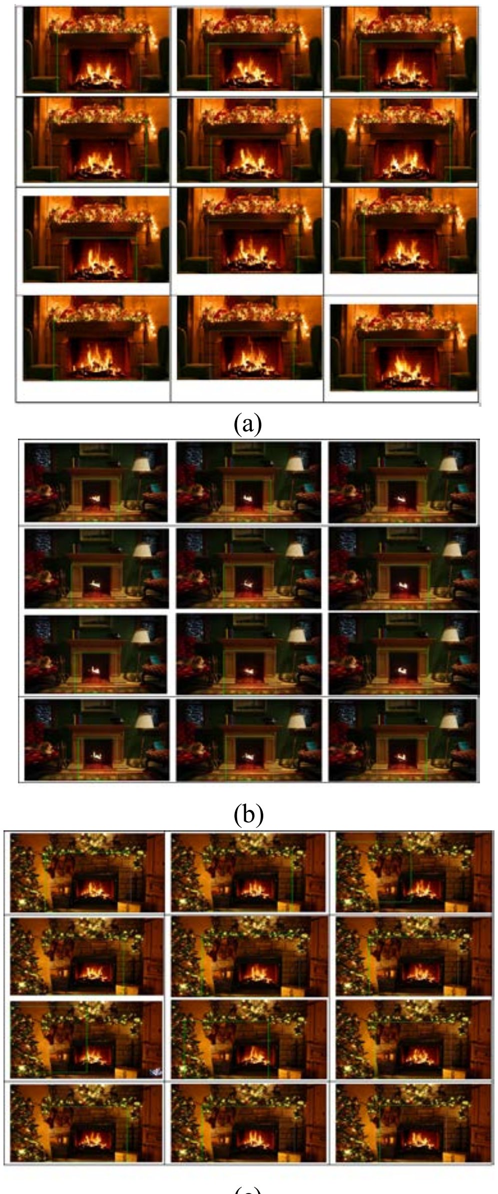
For the training dataset, we have 172 images of fire flames, we used image processing functions such as flipping, adjusting brightness and saturation to create ten samples from each image. These 1720 samples are different because they have minor distinctions at the pixel level and the positions of bounding boxes are also different, which could significantly reduce time cost of the training data. We now have 20 epochs for pre-training and 40 epochs for formal training, which means that the total training dataset has 60 epochs and 10,320 training times.

IV. Results and Analysis

In the flame detection based on YOLO model, the training set is collected which includes 194 pictures of a fire flame. We used data enhancement, which includes picture rotation, flipping, and bright adjustment to increase the amount of our training data.

A. Results

The YOLO model has a unique way to extract visual features for flame detection. The detected results are shown in Fig. 3. Fig. 3(a), (b) and (c) show three different places for flame detection.

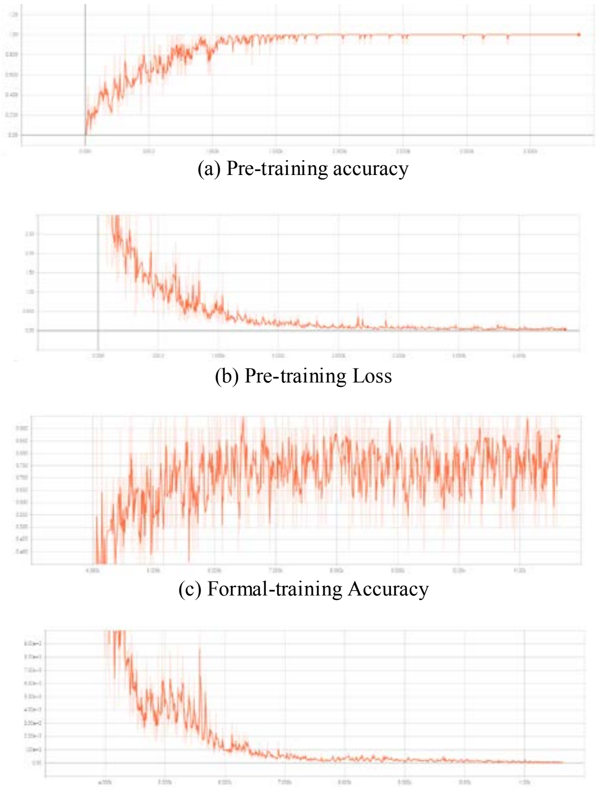

(a)

(b)

(c)

Figure 3. Motion pictures of various videos for flame detection using CNN.

In Fig. 3, the center of a flame region has been detected. From these results, the detected regions are very accurate in most of these frames; but when there are lots of bright objects in the scene, the bounding box may shift a bit from the exact location.



(a) Pre-training accuracy

(b) Pre-training Loss

(c) Formal-training Accuracy

(d) Formal-training Loss

Figure 4. The detection accuracy and loss in pre-training and formal training.

In Fig. 4, after 2,000 times of training, the accuracy of pre-training in Fig.4 (a) is extremely close to 100% and the accuracy of formal training in Fig.4 (c) tends to 76% after 6,000 times training. Figure 4 (c) and (d) show the results of loss in pre-training and formal training.

B. Evaluations of Deep Learning Method

In deep learning, the training time indeed depends on the performance of the devices. When using GPU to train the dataset, it could take approximately 50 times faster than using CPU.

TABLE I. COMPARING TIME COSTS BY USING CPU TO GPU

	Training Time Cost in using CPU and GPU		
	Samples	*Epochs*	*Time Costs*
CPU	1720	2	12 hours
GPU	1720	6	90 minutes

In Fig. 3(a) and (b), the objects in these scenes, which reduce the accuracy of object detection, are the light bulbs above the fireplace. Even if the scenes of Fig. 3(a) and (b) are pretty similar, the result of Fig. 3 (a) is much better than Fig.3 (b). The scene of Fig. 3 (b) is more complicate than Fig. 3 (a) and the left light decreases the accuracy of fire flame detection.

Meanwhile, when the background is relatively simple, the accuracy would perform much better as shown in Fig. 3 (c). In Fig. 3 (c), there is one light on the right side; the light could not affect the detection too much. For evaluating all demo video, over 80% of bounding boxes get the right location of the flames.

When compared to shallow learning using the color-based model, deep learning exhibits its outperformance. When fire flames have entirely different color features compared to the training set, shallow learning may have difficulties to detect them. However, deep learning demonstrates its superior performance in this case; it is not influenced by the changes of flames, thanks to its merits of the fine-grained adaptivity.

C. Limitations

There are obvious limitations of traditional flame detection method; e.g., it requires the developers to have specific domain knowledge of fire flames, such as motion, colors and patterns. With the domain knowledge to set up one better model, it could get higher performance on accuracy. However, most of the time, we do not have such relevant knowledge and this may significantly influence the creation of a specific model. Even if we have the relevant knowledge of this domain, when the object changes, the models need to be rebuilt for the new objects.

Deep learning requires the knowledge from machine learning and artificial neural networks. However, it does not require additional knowledge if the training datasets possess the target object. The model does not need to be changed in order to detect new objects for the adaptively. Also, it could be used for multiple object detection.

In this paper, an optimized YOLO model is proposed. YOLO has an excellent performance on decreasing training time. Meanwhile, because of setting up the 7×7 grid cell, the bounding boxes do not catch the exact region of fire flames; hence, there is still a room for improving this model.

V. CONCLUSION AND FUTURE WORK

There are two primary methods which have been applied to identify fire flames from the viewpoint of digital images and videos: shallow learning and deep learning. Both methods performed well on flame detection; deep learning has a better performance than that of the shallow learning model if the train dataset is big enough.

With an extensive training dataset of fire flames, the performance of deep learning is not affected by using flame colors. However, using the color-based model, a gas fire may not be detected correctly. On the other hand, our method of flame detection using deep learning in this paper only takes YOLO into consideration. There are other deep learning methods, such as autoencoder, sparse coding, restricted

Boltzmann machine, deep belief networks and recurrent neural networks, etc. We could combine them together to develop a better approach for fire flame detection (Ba, Mnih & Kavukcuoglu, 2014).

After thoroughly analyzing the pros and cons of our proposed method, our possible future work includes: (1) we will work on improving the YOLO model to increase the accuracy of flame detection. Other CNN models (e.g., Faster RCNN) could have a better performance on accuracy even if they could take more time to train. (2) The combination of YOLO and Faster RCNN [31] models together will have a better performance than using YOLO only but it will cost more training time [32]. (3) By using deep learning model, not only the flame detection but also other object detection would be right in applications if the deep learning models could be improved.

REFERENCES

[1] Celik, T. Fast and efficient method for fire detection using image processing. ETRI, 32(6), 881-890, 2010.

[2] Marbach, G., Loepfe, M., & Brupbacher, T. An image processing technique for fire detection in video images. Fire safety, 41(4), 285-289, 2006.

[3] Vicente, J., & Guillemant, P. An image processing technique for automatically detecting forest fire. International Journal of Thermal Sciences, 41(12), 1113-1120, 2002.

[4] Toulouse, T., Rossi, L., Celik, T., & Akhloufi, M. Automatic fire pixel detection using image processing: a comparative analysis of rule-based and machine learning-based methods. Signal, Image and Video Processing, 10(4), 647-654, 2016.

[5] Wang, L., Li, A., Yao, X., & Zou, K. Fire Detection in Video Using Fuzzy Pattern Recognition. International Conference on Oriental Thinking and Fuzzy Logic, 2016, pp. 117-127. Springer International Publishing.

[6] Liu, C. B., & Ahuja, N. Vision based fire detection. International Conference on Pattern Recognition, 2004, Vol. 4, pp. 134-137.

[7] Toreyin, B. U., Dedeoglu, Y., & Cetin, A. E. Flame detection in video using hidden markov models. IEEE International Conference on Image Processing, 2005, Vol. 2, pp. II-1230.

[8] Celik, T., Ozkaramanlı, H., & Demirel, H. Fire and smoke detection without sensors: image processing based approach. European Signal Processing Conference, 2007, pp. 1794-1798.

[9] Hanamaraddi, P. M. A Literature Study on Image Processing for Forest Fire Detection. IJITR, 4(1), 2695-2700, 2016.

[10] Toreyin, B. U., Dedeoglu, Y., Gudukbay, U., & Cetin, A. E. Computer vision based method for real-time fire and flame detection. Pattern recognition letters, 27(1), 49-58, 2006.

[11] Jun, C., Yang, D., & Dong, W. An early fire image detection and detection algorithm based on DFBIR model. World Congress on Computer Science and Information Engineering, 2009, Vol. 3, pp. 229-232.

[12] LeCun, Y., Bengio, Y., & Hinton, G. Deep learning. Nature, 521(7553), 436-444, 2015.

[13] Celik, T., Demirel, H., Ozkaramanli, H., & Uyguroglu, M. Fire detection using statistical color model in video sequences. Journal of Visual Communication and Image Representation, 18(2), 176-185, 2007.

[14] Hecht-Nielsen, R. Theory of the backpropagation neural network. Neural Networks, 1, 445-448, 1988.

[15] Wan, J., Wang, D., Hoi, S. C. H., Wu, P., Zhu, J., Zhang, Y., & Li, J. Deep learning for content-based image retrieval: A comprehensive study. ACM Multimedia, 2014, pp. 157-166.

[16] Chan, T. H., Jia, K., Gao, S., Lu, J., Zeng, Z., & Ma, Y. Pcanet: A simple deep learning baseline for image classification. IEEE Transactions on Image Processing, 24(12), 5017-5032, 2015.

[17] Zeng, D., Liu, K., Lai, S., Zhou, G., & Zhao, J. Relation Classification via Convolutional Deep Neural Network. COLING, 2014, pp. 2335-2344.

[18] Ciresan, D. C., Meier, U., Gambardella, L. M., & Schmidhuber, J. Convolutional neural network committees for handwritten character classification. International Conference on Document Analysis and Recognition (ICDAR), 2011, pp. 1135-1139.

[19] Kim, Y. Convolutional neural networks for sentence classification. International Conference on Empirical Methods in Natural Language Processing, 2014, pp. 1746-1751.

[20] LeCun, Y., & Bengio, Y. Convolutional networks for images, speech, and time series. The handbook of brain theory and neural networks, 3361(10), 1995.

[21] Lawrence, S., Giles, C. L., Tsoi, A. C., & Back, A. D. Face recognition: A convolutional neural-network approach. IEEE transactions on neural networks, 8(1), 98-113, 1997.

[22] Krizhevsky, A., Sutskever, I., & Hinton, G. E. Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems, 2012, pp. 1097-1105.

[23] Matsugu, M., Mori, K., Mitari, Y., & Kaneda, Y. Subject independent facial expression recognition with robust face detection using a convolutional neural network. Neural Networks, 16(5), 555-559, 2003.

[24] Dahl, G. E., Sainath, T. N., & Hinton, G. E. Improving deep neural networks for LVCSR using rectified linear units and dropout. IEEE International Conference on Acoustics, Speech and Signal Processing, 2013, pp. 8609-8613.

[25] Sun, Y., Wang, X., & Tang, X. Deep learning face representation from predicting 10,000 classes. IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 1891-1898.

[26] Giusti, A., Ciresan, D. C., Masci, J., Gambardella, L. M., & Schmidhuber, J. Fast image scanning with deep max-pooling convolutional neural networks. IEEE International Conference on Image Processing, 2013, pp. 4034-4038.

[27] He, K., Zhang, X., Ren, S., & Sun, J. Spatial pyramid pooling in deep convolutional networks for visual recognition. European Conference on Computer Vision, 2014, pp. 346-361. Springer International Publishing.

[28] Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., & Darrell, T. Caffe: Convolutional architecture for fast feature embedding. ACM Multimedia, 2014, pp. 675-678.

[29] Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., ... & Kudlur, M. TensorFlow: A system for large-scale machine learning. USENIX Symposium on Operating Systems Design and Implementation (OSDI), 2016, Savannah, Georgia, USA.

[30] Bergstra, J., Breuleux, O., Bastien, F., Lamblin, P., Pascanu, R., Desjardins, G., & Bengio, Y. Theano: A CPU and GPU math compiler in Python. 9th Python in Science Conference, 2010, pp. 1-7.

[31] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. You only look once: Unified, real-time object detection. IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 779-788.

[32] S. Ren, K. He, R. Girshick, J. Sun. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. IEEE Transactions on Pattern Analysis and Machine Intelligence, 39(6), 1137 - 1149, 2017.