Using Popular Object Detection Methods for Real Time Forest Fire Detection

Shixiao Wu
College of Information Engineering
Wuhan Business University
Wuhan, China
e-mail: 343564602@qq.com

Libing Zhang
Department of Organization
Wuhan Business University
Wuhan, China
e-mail: 1412860232@qq.com

Abstract*—In this paper, we focus on three problems that surrounded forest fire detection, real-time, early fire detection, and false detection. For the first time, we use classical objective detection methods to detect forest fire: Faster R-CNN, YOLO (tiny-yolo-voc, tiny-yolo-voc1, yolo-voc.2.0, and yolov3), and SSD, among them SSD has better real-time property, higher detection accuracy and early fire detection ability. We make the fire and smoke benchmark, utilize the new added smoke class and fire area changes to minimize the wrong detection. Meanwhile, we adjust YOLO's tiny-yolo-voc structure and propose a new structure tiny-yolo-voc1, the experiments proves that this improves the fire detection accuracy rate. This paper is very practical for forest safety and real time forest monitor.

Keywords-component; real time forest fire detection; object detection; forest safety; improved YOLO

I. INTRODUCTION

Various fire detection methods appeared out, human observation, Satellite Systems, IR, WSN, visual/image based techniques and so on[1]. Human observation is one of the oldest and traditional methods, labor-consuming and time-consuming. Satellite systems need a long scan period and cannot provide a real-time fire image. IR may cause scatter of the transmitted beam. Although many people focus on the research of WSN fire detection, they have to resolve the difficulties of how to distribute the sensors in complex outdoor environments and battery charge. Compared with these methods, visual/image based methods exhibit bigger advantages. They can monitor forest 24-hours and detect fire as early as possible. Visual/image based techniques always detect three aspects of the forest fire, color, texture, motion. The reality is although a lot of literatures has higher detection accuracy and lower false detection, real-time is often ignored.

In this paper, we evaluate the experiments through three popular object detection methods, Faster R-CNN[2], YOLO(tiny-yolo-voc, tiny-yolo-voc1, yolo-voc 2.0, yolov3)[3] and SSD[4]. We compare the detection accuracy and train time, finally find that Faster R-CNN has shorter train time and the higher detection accuracy. But as Shaoqing Ren mentioned in the paper, with a K40 GPU, Faster R-CNN has a frame-rate 17 fps with the ZF net. Using NVIDIA 1070, we have trained YOLO (tiny-yolo-voc) and finally get a frame-rate between 15fps and 25fps. We believe that if we change the faster GPU, the fps will be higher. So speaking of the real-time detection of Faster R-CNN, we only compare the performance of YOLO and SSD, the final results prove that SSD has better detection accuracy and faster frame-rate.

YOLO is famous for its real-time property, but for small object it always has lower detection accuracy rate. We improve this by adding two more layers into tiny-yolo-voc. The experiments shows that this change improve the detection accuracy rate, the improved tiny-yolo-voc is even better than yolo-voc.2.0 (it has more layers than tiny-yolo-voc). The detection accuracy of this new structure is lower than YOLOv3 however. Make an early fire detection is also very important, we test a lot of smaller cooler fire and find that SSD still perform better than YOLO.

Except that, fire-colored ordinary objects are wrongly detected as fire happen frequently, we add another class smoke to enhance the reliability of fire detection. Considering fire will be bigger/smaller, we extract some key frames from real-time video detection and compare the area change between prior fire frame and next fire frame, if the area becomes bigger/smaller, that is real fire, OpenCV helps us to finish this job. We conclude our contributions as follows:

For the first time, we make the fire/smoke detection benchmark and use three popular object detection methods to detect real-time forest fire. We believe that SSD achieves the best performance. SSD has higher fps, not only detect early smaller fire but also has the higher detection accuracy. We think that these three methods is very practical for forest fire detection, Faster R-CNN has higher detection accuracy, YOLO is faster and if train

2473-3547/18/$31.00 ©2018 IEEE
DOI 10.1109/ISCID.2018.00070

time is longer, the final results will be very good, SSD is accurate and fast.

To YOLO, we alter the YOLO's tiny-yolo-voc and get a better fire detection accuracy. We use 100 test images to compare the performance of 4 YOLO structures (original tiny-yolo-voc, tiny-yolo-voc1, yolo-voc.2.0, and yolov3), yolov3 performs better. Tiny-yolo-voc1 is better than tiny-yolo-voc and yolo-voc.2.0.

We not only consider fire detection but also smoke detection. Smoke class is added to enhance the reliability of the fire. Further, we use area changes detection to distinguish the wrongly detection between the real fire and fire-colored objects. The changed area also helps us to know the area of fire disaster.

II. Related Work

With the help of clustering and fuzzy logic, Kalli Srinivasa Nageswara Prasad et al. [5] have proposed a novel scheme to automatically detect forest fire from the spatial data corresponding to forest regions. A formed fuzzy rule comprises of four steps, color space conversion, K-means clustering, fuzzy set generation and fuzzy rules derivation. With the aid of publicly available spatial data, the formed fuzzy rules have efficiently detected the fires.

Punam Patel et al. [6] combines color detection, area dispersion and motion detection to detect fires in video frames. RGB is taken to detect red color information from images, and then color space transformation equation is used to generate a corresponding Y, Cb, Cr image. They use Frame differencing method to subtract out extraneous background noise to detect moving pixels in video images. They analyze two sequential frames and check out dispersion in coordinate (minimum and maximum) of X and Y, then compare to have model of area detection.

By identifying gray cycle pixels nearby the flame, Gaurav Yadav et al. [7] give optimization on fire detection scheme. Based on color detection, they proposed a novel fire detection system that including motion detection, gray cycle detection and area dispersion, finally the system performance is 92.31%. They believe that this system employs less false alarm and higher system performance. Sam G.Benjaminetal., [8] concludes different techniques that drastically reduce the false detection rate. These authors consider that mulitiple techniques combination is essential to obtain better detection results. They prove that techniques conclude color clues, motion analysis and fire flickering perform better than sticking onto color information alone.

Anupam Mittal et al. [9] give review of machine learning techniques for fire detection. SVM, ANN, DT, FFNN are introduced to readers. For forest fire detection using SPOT-4 imagery, F. Sunar et.al [10] investigate the capabilities of boosting classification approach. Five classification techniques include Multi Layer Perception (MLP), Maximum Likelihood (ML), Adaboost (AB), Logitboost (LB) and Regression Tree (RT) are assessed through classification accuracy. The result shows that AB and LB classifications could be a great potential alternative to conventional techniques.

Qingjie Zhang et al. [11] provide a deep learning method for forest fire detection. They operated the fire detection in a cascaded fashion, first the global image-level test the full image and then the fine gained patch classifier detect the precise location of the detected fire. They proposed a fire detection benchmark, 178 images for train set and 59 images for test set. For the first stage, they adopt the CIFAR 10 network, but the number of output is changed by 2, also they add a drop out layer for avoiding overfitting. For the second stage, they use 8 layers AlexNet available in Caffe framework.

Our paper focus on real-time forest fire/smoke detection, we use 3 mentioned methods and one improved YOLO algorithms to carry out the experiments. The results shows that SSD has the best performance and the best real-time property. We improve the original tiny-yolo-voc and get a new structure tiny-yolo-voc1, which shows better detection accuracy than yolo-voc.2.0 and tiny-yolo-voc. We focus on small/dark fire/smoke real-time detection, although improved structure of YOLO has real-time property, but bad performance in small fire make us abandon it and adopt SSD.

III. Methods

A. Faster R-CNN

Before Faster R-CNN come out, region proposal computation is a bottleneck for object detection networks at that time. Faster R-CNN introduce Region Proposal Network(RPN) to solve that1problem, this proposed network shares convolutional features with Fast R-CNN and enables cost-free region proposals. Because they have a low frame rate of 5fps on a GPU (k40 and ZF is 17 fps), we give up to test real-time on Faster R-CNN, just discuss static pictures detection.

They add two additional convolutional layers to construct RPNs, one is responsible for encoding each convolutional feature map position into a short feature

vector, and the other one generates an objectness score and regressed bounds for k region proposals. By the last shared convolutional layer, they slide a small network over the convolutional feature map output. To an n*n spatial window of the input convolutional feature map, this small network is fully connected. Take ZF for 160 example, each sliding window is mapped to a 256-d vector, then this vector will be fed into two full-connected convolutional layers, a box-regression layer(reg) and a box-classification layer(cls).The cls layer that estimate probability of object/not-object for each proposal will output 2k scores, the reg layer that encodes the coordinates of k boxes has 4k outputs, and each sliding-window has k region proposals (anchors).

Through alternating optimization, they learn shared features by developing a pragmatic 4-step training algorithm. Sharing convolutional layers occurs only at the third step, at that time they fine-tune the layers unique to RPN and fix the shared convolutional layers.

As Shaoqing Ren mentioned in the paper, with a K40 GPU, Faster R-CNN has a frame-rate 17 fps with the ZF net. Our GPU is NVIDIA 1060, cannot reach the performance that the paper mentioned, so we just talk about static performance and ignore the real-time test. We believe that if the GPU is very well, Faster R-CNN can have a very good performance.

B. YOLOv3

YOLOv3 use dimension clusters as anchor boxes to predict bounding Boxes, K-means is adopted to generate 9 clusters and determine bounding box priors [11]. The clusters is divided up evenly across 3 different scales, for that YOLOv3 has good APs performance. For each bounding boxes, this network predicts 4 coordinates, which indicate the location information, width and height. Logistic regression is utilized to generate an objectness score. Considering that a multilabel approach better models the data, they adopt independent logistic classifiers.

YOLOv3 is a hybrid network, successive 3*3 convolutional layers, 1*1convolutional layers and some shortcut connections are included. This network has 53 convolutional layers.

C. SSD

SSD is a single-shot detector which not only eliminates the bounding box proposals but also gives up the pixel or feature resampling stage. For a fixed set of default bounding boxes, they utilize small convolutional filters that applied to feature maps to predict category scores. To perform detection at multiple scales, they apply separate filters for different aspect ratio detections from the later stage of a network.

SSD model make use of the VGG-16 as a base and except that, some convolutional feature layers are added to the end of the truncated base network. These convolutional layers allow multiple scale detection and decrease in size progressively. For predicting parameters of a potential detection, the basic element is a 3*3*p small kernel, this kernel will produces either a shape offset relative to the default box coordinates or a score for a category. This kernel will produce an output value whenever it's applied. The developers apply default boxes that are associated with each feature map cell to predict the offsets and per-class score. For SSD, two fully connected layer is abandoned and convolutional layers is used to predict the output value.

The training stage of SSD involves hard negative mining, data augmentation and the set of default boxes and scales for detection selection. With the best jaccard overlap, the default box is matched to each ground truth box.

IV. EXPERIMENTS

In this section, we will show experiments results through 3 object detection methods, Faster R-CNN, YOLO (tiny-yolo-voc1, tiny-yolo-voc, yolo-voc.2.0, and yolov3) and SSD.

For Faster R-CNN, we give a result based on 120000 iteration times. For YOLO, we find that YOLO has a bad performance on smaller cooler fire detection, so we try to adjust the structure of tiny-yolo-voc by adding two more layers (one is convolutional layer and the other is maxpooling layer, the filter of convolutional layer is 8). When training is finished, we finally find that these two added layer boost the original smaller fire detection accuracy rate. The experiment result proves that more layer with small filter catch more details. For SSD, we test its static and real-time detection accuracy rate on smaller fire, the result shows that this methods has better performance than YOLO (tiny-yolo-voc), it can make an accurate and real-time detection an smaller fire

A. Faster R-CNN

We use 1000 fire pictures with 300*300 size as bench mark. BBox-Label-Tool v2 is used to label the pictures, fire and smoke are both labeled therein. We alter the iterations for each training stage at 120000, 60000, 120000, and 60000 and keep the default values for all the other settings.

Fire and smoke detection accuracy rate both are very good, even the very small fire Faster R-CNN can detect rightly. For smoke detection accuracy rate, Faster R-CNN is close to 1. For fire detection, detection accuracy rate for the small fire is 0.99 and the dark fire is 0.974. We only show static picture herein, because as the author of SSD said, it can only operates at only 7 frames per second (FPS), too slow for real-time fire-detection applications. We focus on research on YOLO and SSD, the later has higher FPS and can satisfy the real-time detection need. Table I describes performance for Faster R-CNN.

TABLE I. PERFORMANCE FOR FASTER R-CNN

Iteration times	Training images	Testing images	Fire accuracy	Smoke accuracy	Net
120000	1000	100	99.7%	79.7%	zf

Figure 1. Performance for Faster R-CNN.

Figure 1 describes performance for Faster R-CNN. The detection accuracy is very accurate, very close to 1. It can detect small fire precisely.

B. YOLO

For YOLO, we test how different layers structure influence the accuracy rate. Still, we use 1000 pictures to make the fire/smoke datasets, the same datasets for those three object detection methods. We use the tiny-yolo-voc structure to train the datasets, finally find that when the iteration times equals to 120000 and learning rate is 0.01, this original structure has plain accuracy. We adjust the tiny-yolo-voc by adding two layers, which include 1 convolutional layer(the filter is 8) and 1 maxpooling layer, the results proves that this new structure improves both fire and smoke detection accuracy. Even we train the yolo-voc.2.0, this new structure still shows better performance. But in the end, yolov3 performs better.

When we train the original tiny-yolo-voc, we initialize the images size as 416*416, during the experiments, we find that when the image size is set as 608*608, the performances become better. So next when we train yolo-voc.2.0 and tiny-yolo-voc1, we both initialize the image size at 608*608. When we utilize tiny-yolo-voc1 to train fire only, it performs best. But when the new class smoke is added, the fire accuracy decrease 10%. When we train dataset using yolov3, we finally find that this configuration performs best.

Herein single training means fire training, combine training means fire/smoke joint training. Figure 2 shows performance for YOLOv3. YOLOv3 has a bad performance for small fire.

Figure 2. YOLOv3 detection accuracy (Left: smoke 98%, fire 100%. Right: fire 65%, smoke 30%).

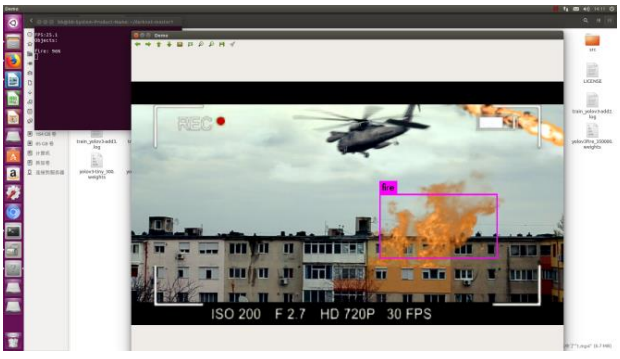

Figure 3. YOLO real-time detection(Fire detection accuracy :96%, FPS:25.1)

TABLE II. PERFORMANCE FOR YOLO MODEL

Iteration times	Training images	Testing images	Fire accuracy	Smoke Accuracy	Net	Images size	Learning rate
120000	1000	100	44.59%		tiny-yolo-voc	416*416	0.01
120000	1000	100	87.01%		tiny-yolo-voc1	608*608	0.01
120000	1000	100	75.12%	71.3%	tiny-yolo-voc1	608*608	0.01
120000	1000	100	63.05%	63.39%	yolo-voc.2.0	608*608	0.01
120000	1000	100	92.29%	65.53%	yolov3	416*416	0.0001

C. SSD

In this section, we use SSD 300*300 model to train our fire datasets. The resized width and resize height are both 300, the batch size equals to 1, number of the testing images equals to 342 and number of the training image is 668, the iteration times is 120000. When the training ends, we finally find that the fire detection accuracy for 100 images is up to 1, except one very small fire which equals to 0.58. For smoke detection, accuracy of in 100 test images are 0, but 13 in 27 are no smoke images, so the missing detection rate is 14 percent. The smoke accuracy rate of these remain 73 images are up to 97.88 percent.

TABLE III. USING SSD FOR FOREST FIRE/SMOKE DETECTION

Iteration times	Training images	Testing images	Fire accuracy	Smoke accuracy	Net
120000	668	342	99.88%	68.4%	zf

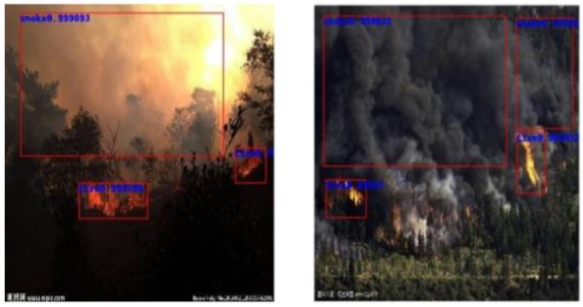

Figure 4. Performance for SSD (fire/smoke detection accuracy is close to 1, very precisely).

Figure 5. Real-time Fire/smoke detection accuracy for SSD (For the left, FPS 27.07, smoke 100%, fire 100%, for the right, smoke 100%, fire 100% FPS 29.15)

D. Area changes

For false detection, we think area changes detection helps a lot. When the fire is bigger and bigger, the area of the fire is growing. The SSD can detect the area of the fire by four value, Xmin, Ymin, Xmax and Ymax. These four values present the coordinates of the top left corner and the coordinates of the lower right corner, then we calculate the area very easily. We catch the two interval frame of the fire usually, when the area grows bigger, this must be the fire.

ACKNOWLEDGMENT

This paper is supported by Wuhan Business University teaching and research project: 2017Y018.

V. REFERENCES

[1] Kaur, A.; Sethi, R.; Kaur, K. Comparison of Forest Fire Detection Techniques Using WSNs. International Journal of Computer Science & Information Technolo 2014.

[2] Ren, S.; Girshick, R.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. IEEE Transactions on Pattern Analysis & Machine Intelligence 2017,235 39, 1137–1149.

[3] Redmon, J.; Farhadi, A. YOLO9000: Better, Faster, Stronger 2016.

[4] Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. SSD: Single Shot MultiBox Detector 2015. pp. 21–37.

[5] Prasad, S.N.; Ramakrishna, S. An Autonomous Forest Fire Detection System Based On Spatial Data Mining and Fuzzy Logic. International Journal of Computer Science & Network Security 2008, pp. 49–55.

[6] Patel, P.; Tiwari, S. Flame Detection using Image Processing Techniques. International Journal of Computer Applications 2012, 58, 45–50.

[7] Yadav, G.; Gupta, V.; Gaur, V.; Bhattacharya, M. OPTIMIZED FLAME DETECTION USING IMAGE PROCESSING BASED TECHNIQUES. Indian Journal of Computer Science & Engineering 2012, 3.

[8] Sam G. Benjamin, R.B. A Comparative Analysis on Different Image Processing Techniques for Forest Fire Detection. International Journal of Computer Science and Network 2016.

[9] Anupam Mittal, Geetika Sharma, R.A. Forest Fire Detection Through Various Machine Learning Techniques using Mobile Agent in WSN. International Research Journal of Engineering and Technology 2016.

[10] Management, Education and Information Technology Application, 2016.

[11] Redmon J, Farhadi A. YOLOv3: An Incremental Improvement[J]. 2018.