Research & Reviews: Journal of Statistics and Mathematical Sciences

Using Combination Methods To Improve Real Time Forest Fire Detection

Shixiao Wu[1,2]* and Chengcheng Guo[1]

[1]College of Electronical and Information, Wuhan University, Wuhan, China

[2]Department of Information Engineering, Wuhan Business University, Wuhan, China

Research Article

Received date: 04/12/2018

Accepted date: 20/12/2018

Published date: 01/01/2019

***For Correspondence**

College of Electronical and Information, Wuhan University, Wuhan, China.

E-mail: 343564602@qq.com

Keywords: Real time forest fire detection, Objective detection, Forest safety, PCA, YOLOv3, SSD

ABSTRACT

This study investigated the potential for using principal component analysis (PCA) to improve real-time forest fire detection with popular algorithms, such as YOLOv3 and SSD. Before YOLOv3/SSD training, we utilize PCA to extract features. Results showed that PCA with YOLOv3 increased the mean average precision (mAP) and the detection accuracy by 3.3% and 16.3% separately. PCA with SSD increased mAP and detection accuracy by 1% and 2.1% separately. These results suggest PCA to be a robust tool for improving different objective detection networks. This paper is very practical for forest safety and real time forest monitor.

INTRODUCTION

Various fire detection appeared out, human observation, Satellite Systems, IR, WSN, visual/image based techniques and so on [1,2]. 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 scattering 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 literature has a higher detection accuracy and lower false detection, real-time is often ignored.

Popular object detection methods like Faster R-CNN, YOLOv3 and SSD can detect forest fire real-time. Tsung-Yi Lin et.al, consider that current state-of-the-art object detectors can be classified as two-stage detectors and one-stage detector. Take the R-CNN framework, for example, this two-stage detector first generates a sparse set of candidate object locations and then implement classifier jobs. Two-stage detectors win the challenging COCO benchmark a lot of times. But recent work on one-stage detector also shows promising results. Compared with state-of-the-art two-stage methods, they yield faster detectors with accuracy within 10~40% [3,4].

This paper pushes the one-stage detector further: we first adopt PCA to pre-process the original color pictures of the forest fire, then throw these processed pictures into the training network. We compare the final detection result of YOLOv3, PCA with YOLOv3, SSD, and PCA with SSD, finally find that the combination methods (PCA with YOLOv3/PCA with SSD) perform better than the individual methods. The mAP and the detection accuracy of the combination methods rise, they get better location result.

For PCA with YOLOv3, we extract 260 features from the original forest fire color images. For PCA with SSD, we extract 300 features from the original ones. We define the features from a lot of tests. The motivation of this proposed job is that we consider that only primary features contribute to the final detection, other redundant features/information from the original pictures make the detection worse. In the preprocessing job of PCA, SVD is adopted to choose the fit ones. Experiments show that our proposed combination methods enable us to train a higher- accuracy/ mAP, one stage detector that significantly outperforms the individual ones.

To demonstrate the effectiveness of the proposed combination methods (PCA with YOLOv3, PCA with SSD), we adopt 3 metrics to evaluate. The detection accuracy, mAP and the training time (10000 times) are listed in the table to show these four methods.

RELATED WORK

With the help of clustering and fuzzy logic, Kalli Srinivasa Nageswara Prasad et al. have proposed a novel scheme to automatically detect forest fire from the spatial data corresponding to forest regions [5]. A formed fuzzy rule comprises 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. combines color detection, area dispersion and motion detection to detect fires in video frames [6]. RGB is taken to detect red color information from images, and the 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 the coordinate (minimum and maximum) of X and Y, then compare to have a model of area detection.

By identifying gray cycle pixels nearby the flame, Gaurav Yadav et al. give optimization on fire detection scheme [7]. 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., concludes different techniques that drastically reduce the false detection rate [8]. These authors consider that multiple 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. give a review of machine learning techniques for fire detection [9]. SVM, ANN, DT, FFNN are introduced to readers. For forest fire detection using SPOT-4 imagery, F. Sunar et.al investigate the capabilities of boosting classification approach [10]. Five classification techniques include Multi-Layer Perceptron (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 previous techniques.

Qingjie Zhang et al., provides a deep learning method for forest fire detection [11]. 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 a train set and 59 images for the 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.

All these previous works does not mention real-time forest fire detection. Real-time is needed for forest fire detection. When the danger comes, timely find can rescue many lives and properties. The proposed two methods (PCA with YOLOv3, PCA with SSD) can detect forest fire real-time. PCA pre-process the original color images before training, and we utilize the trained weights/ caffemodel to detect the forest fire. This preprocessing not only does not affect the final real-time detection but also increase the detection accuracy and mAP.

METHODS

YOLOv3

YOLOv3 uses dimensional clusters as anchor boxes for predicting bounding boxes. The K- means algorithm was adopted in this study to generate 9 clusters and determine bounding box priors [11]. These clusters were then divided evenly across 3 different scales (13 × 13, 26 × 26, and 52 × 52), resulting in good average precision (AP) performance with YOLOv3. Several convolutional layers were included in the base feature extractor, the last of which predicted 3D tensor encoding objectivity, bounding box structure, and class. The network then predicted 4 coordinates for each bounding box, indicating location coordinates, width, and height. Logistic regression was utilized to generate an object score, which measured the overlap between the bounding box and a ground truth object. A prediction was made if the bounding box prior overlapped the ground truth object by more than the threshold of 0.5. Independent logistic classifiers, adopted as a multi-label approach, proved to be a better model for the data. As such, we propose that softmax is unnecessary for achieving good detection performance. Class- specific confidence scores were obtained as follows:

$$\Pr(\text{Classi}|\text{Object}) \times \Pr(\text{Object}) \times \text{IOU} = \Pr(\text{Classi}) \times \text{IOU} \quad (1)$$

YOLOv3 is a hybrid network which utilizes successive 3 × 3 convolutional layers, 1 × 1 convolutional layers, and certain shortcut connections which skip one or more layers in a manner similar to a Resnet. The outputs of the shortcut were added to the

outputs of the stacked layers. Our implementation of this convolutional layer network (the original YOLOv3 network, Darknet-53) is shown in **Figures 1 and 2**.

	Type	Filters	Size	Output
	Convolutional	32	3 × 3	256 × 256
	Convolutional	64	3 × 3 / 2	128 × 128
1×	Convolutional	32	1 × 1	
	Convolutional	64	3 × 3	
	Residual			128 × 128
	Convolutional	128	3 × 3 / 2	64 × 64
2×	Convolutional	64	1 × 1	
	Convolutional	128	3 × 3	
	Residual			64 × 64
	Convolutional	256	3 × 3 / 2	32 × 32
8×	Convolutional	128	1 × 1	
	Convolutional	256	3 × 3	
	Residual			32 × 32
	Convolutional	512	3 × 3 / 2	16 × 16
8×	Convolutional	256	1 × 1	
	Convolutional	512	3 × 3	
	Residual			16 × 16
	Convolutional	1024	3 × 3 / 2	8 × 8
4×	Convolutional	512	1 × 1	
	Convolutional	1024	3 × 3	
	Residual			8 × 8
	Avgpool		Global	
	Connected		1000	
	Softmax			

Figure 1. YOLOv3[2].

Figure 2. PCA with YOLOv3 training.

SSD

SSD is a single-shot detector which not only eliminates bounding box proposals but also excludes the pixel or feature resampling stage [12]. For a fixed set of default bounding boxes, the algorithm utilizes small convolutional filters, which are applied to feature maps in predicting category scores. Detection is performed at multiple scales by applying separate filters for different aspect ratios in the latter stages of the network. The SSD model makes use of VGG-16 as a base, with convolutional feature layers added to the end of the truncated base network. These layers allow for multiple-scale detection and successively decrease in size. The basic element used for potential parameter detection was a 3 × 3 × p (see **Figure 3**) small kernel, which produced either a shape offset relative to the default box coordinates or a score for a specific category. This kernel will produce an output value whenever it is used. The applied default boxes associated with each feature map cell are used to predict offsets and per-class scores. **Figure 4** shows the architecture SSD, where two fully-connected layers are discarded and the convolutional layers are reused to predict the output value. The overall objective loss function was a weighted sum of the confidence loss (conf) and the localization loss:

$$L(x,c,l,g)=\frac{1}{N}\left(L_{conf}(x,c)+\alpha L_{loc}(x,l,g)\right) \quad (2)$$

Here N represents the number of matched default boxes. The localization loss is a smooth L1 loss between the ground truth box (g) parameters and the predicted box (l). The width (w) and height (h) of the offsets regress for the center (cx, cy) of the default bounding box (d):

$$L_{loc}(x,l,g)=\sum_{i\in Pos}^{N}\sum_{m\in\{cx,cy,w,h\}}x_{ij}^{k}smooth_{L1}\left(l_i^m-\hat{g}_j^m\right),$$

$$\hat{g}_j^{cx}=\left(g_j^{cx}-d_i^{cx}\right)/d_i^w\,\hat{g}_j^{cy}=\left(g_j^{cy}-d_i^{cy}\right)/d_i^h, \quad (3)$$

$$L_{conf}(x,c) = -\sum_{i\in Pos}^{N} x_{ij}^{P} \log\left(\hat{c}_i^p\right) - \sum_{i\in Neg} \log\left(\hat{c}_i^o\right)$$

$$L_{conf}(x,c) = -\sum_{i\in Pos}^{N} x_{ij}^{P} \log\left(\hat{c}_i^p\right) - \sum_{i\in Neg} \log\left(\hat{c}_i^o\right) \text{ where } \hat{c}_i^p = \frac{\exp\left(c_i^p\right)}{\sum_p \exp\left(c_i^p\right)} \quad (4)$$

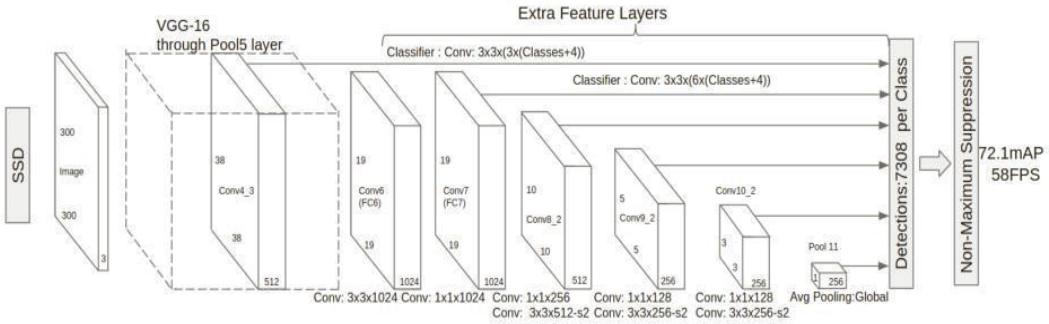

Figure 3. SSD network.

Figure 4. PCA with SSD training procedure.

Principal Component Analysis

Principal Component Analysis (PCA) is a data reduction method similar to Factor Analysis (FA). In this process, a smaller subset of principal components is generated by analyzing the correlation between variables in a data set. Orthogonal transforms are then used to generate principal components, the first of which has the largest possible variance[13]. PCA can be applied using the singular value decomposition of a matrix and can produce a lower-dimensional image, representing a projection of an object when viewed from its most informative viewpoint.

This process is defined for a data matrix X with a size of n × p, wherein p is the number of variables or features and n is the number of samples. If X is a centered data matrix, its covariance can be defined as:

$$C = \frac{1}{n-1} X^T X \quad (5)$$

C is a symmetric matrix which can be diagonalized as:

$$X = UEV^T \quad (6)$$

Here, V is a matrix of eigenvectors and Λ is a diagonal matrix with eigenvalues λ_i. Singular value decomposition can then be used to display X as:

$$X = UEV^T \quad (7)$$

The covariance matrix C can then be rewritten using singular value decomposition:

$$c = \frac{1}{n-1} X^T X = \frac{1}{n-1}\left(U\Sigma V^T\right)^T \left(U\Sigma V^T\right) = \frac{1}{n-1} V\Sigma^T U^T U\Sigma V^T = \frac{1}{n-1} V\left(\Sigma^T\Sigma\right)V^T = V\frac{\Sigma^2}{n-1}V \quad (8)$$

In this equation, the eigenvalues λ_i of the diagonal matrix C exhibit the following relationship with the singular values σ^2_i of the matrix X: $\sigma^2_i=(n-1)\lambda_i$.

The PCA algorithm for color images and extracted features is demonstrated in **Figure 4.** YOLOv3 and SSD were used to extract features from the fire images, identifying 260 and 300 features, respectively. The human eye typically cannot recognize differences for features above 50, as seen in **Figures 5 and 6**.

Figure 5. Using PCA to extract features from a forest fire (1,50, 100, 150, 250, 300, from top left to bottom right).

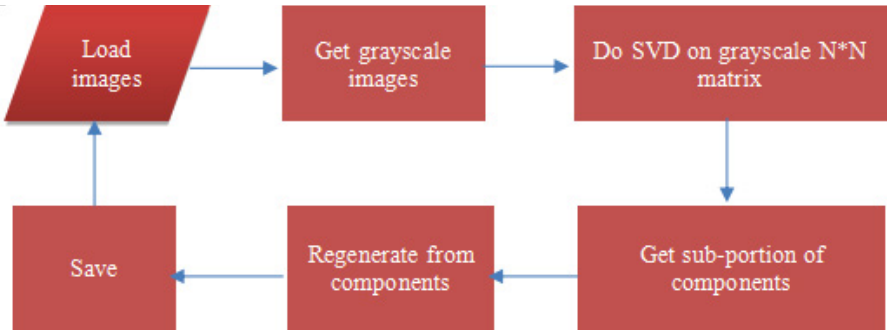

Figure 6. PCA algorithm.

EXPERIMENTAL RESULTS AND ANALYSIS

This section demonstrates the performance of four methods: YOLOv3, YOLOv3 with PCA, SSD, and PCA with SSD. The fire datasets w used for testing purposes. For YOLOv3 with PCA, images were first pre-processed using PCA (N=260) and then used to train. For PCA with SSD, images were first pre-processed using PCA (N=300) and then used to train.

Table 1 illustrates the performance of these four methods. Compared with YOLOv3, PCA with YOLOv3 increased the mAP and the detection accuracy by 4.2% and 16.3%. PCA with SSD increased the mAP and the detection accuracy by 0.4% and 2.1%, the training time are reduced at five minutes. The combination methods show better mAP and detection accuracy.

Herein the detection accuracy means the object score for YOLOv3 and SSD.

Table 1. Performance of four methods: YOLOv3, YOLOv3 with PCA, SSD, and PCA with SSD.

Methods	No. of training images	No. of testing images	Training times (s)	Parameter settings	mAP	Detection accuracy
PCA with YOLOv3	597	150	11734	Learning rate:0.0001 batch=16 subdivisions=4	77.6%	92.4%
YOLOv3	597	150	11631	Learning rate:0.0001 batch=16 subdivisions=4	73.4%	76.1%
SSD	597	150	900	Learning rate:0.0001 Others remain unchanged	87.6%	78.4%
PCA with SSD	597	150	852	Learning rate:0.0001 Others remain unchanged	88%	80.5%

Also, we give the loss curves/IOU curves for PCA with YOLOv3 and YOLOv3 in **Figure 7 and Figure 8**. **Figure 9** illustrates the detection accuracy for PCA with YOLOv3 and YOLOv3. **Figure 10** gives the pictures performance for YOLOv3 and PCA with YOLOv3.

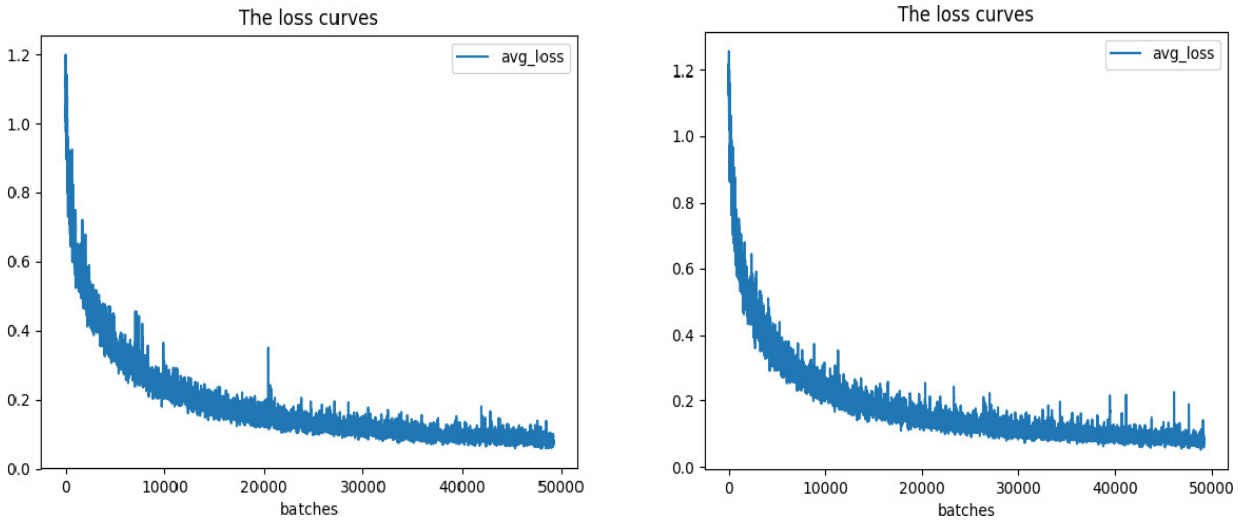

Figure 7. The loss curves for YOLOv3 (left) and PCA with YOLOv3 (right). PCA with YOLOv3 show smoother.

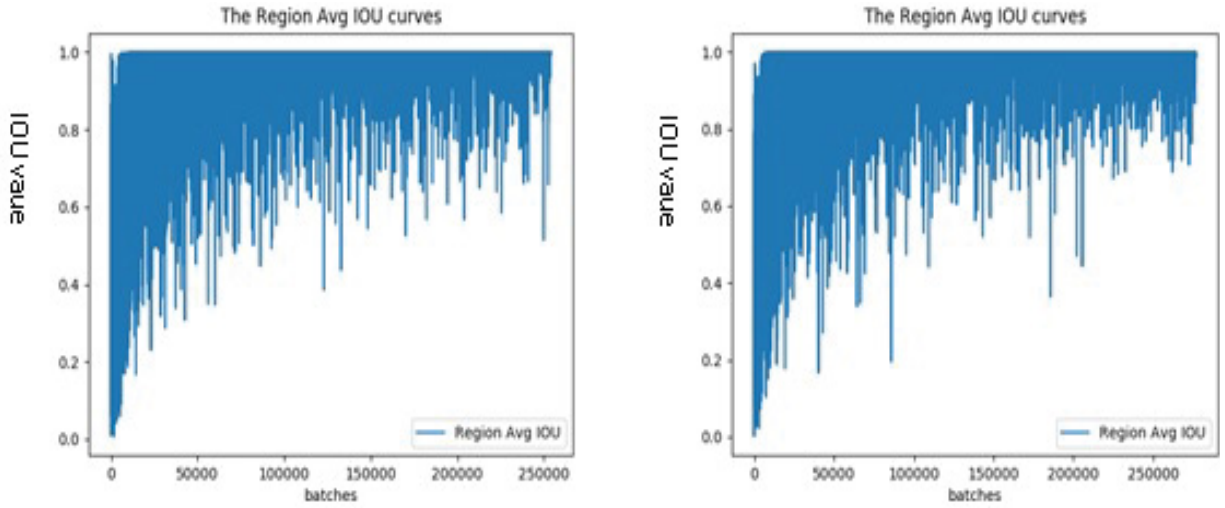

Figure 8. The Regiou Avg IOU curves for PCA with YOLOv3 (left) and YOLOv3 (right). PCA with YOLOv3 show much more higher IOU.

Figure 9. Detection accuracy for PCA with YOLOv3/YOLOv3. PCA with YOLOv3 has higher accuracy.

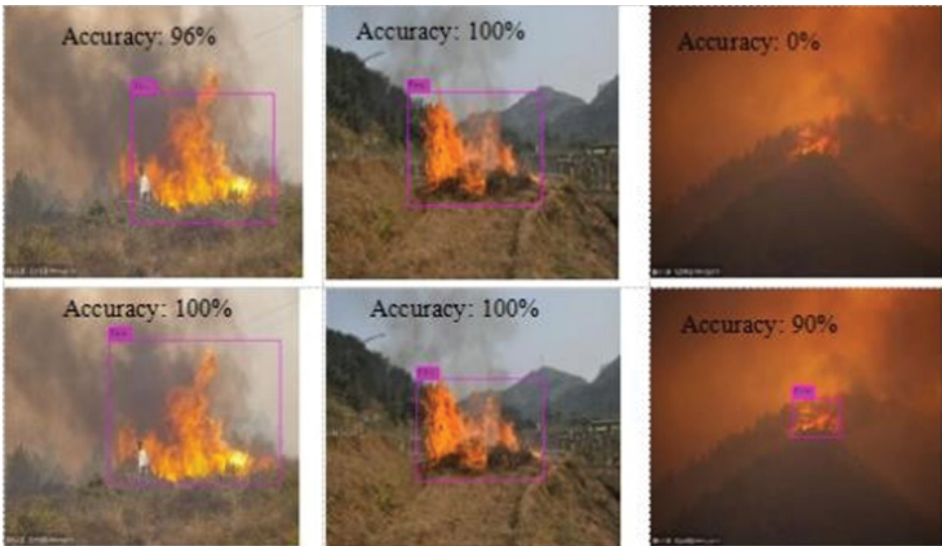

Figure 10. Picture comparison for PCA with YOLOv3 (below) and YOLOv3 (observe). You can observe that PCA with YOLOv3 shows the better location and detection accuracy. The combination method can observe the early fire.

Figure 11 shows training loss for PCA with SSD and SSD. **Figure 12** shows the mAP of PCA with SSD and SSD alone. It has shown mAP at different iteration times. In the figure, we show the mAP as test accuracy. **Figure 13** has shown images comparison for PCA with SSD and SSD alone. **Figure 14** shows the detection accuracy of 150 testing images. Four methods comparison has shown in the figure. **Figure 15** has shown images comparison for four methods.

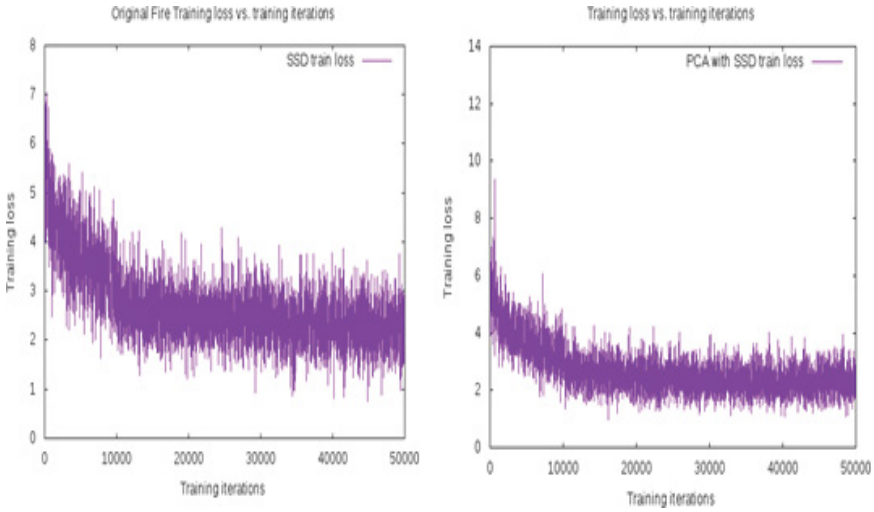

Figure 11. Training loss for SSD (left) and PCA with SSD (right). PCA with SSD shows much more sooth curves.

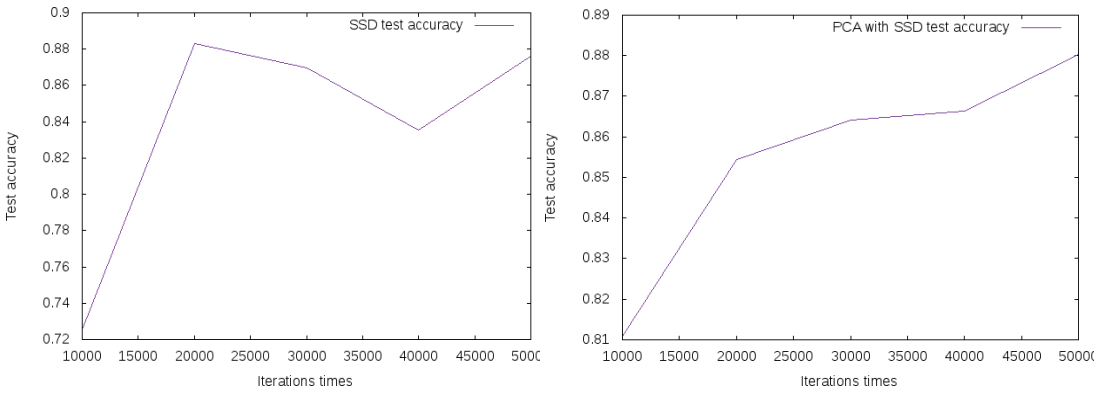

Figure 12. Testing accuracy for SSD (left) and PCA with SSD (right). PCA with SSD shows higher test accuracy. After 50000 iteration times, the test accuracy of PCA with SSD reaches 88. Compared with the combination method, SSD has a little lower test accuracy, 87.6. Here test accuracy means mAP.

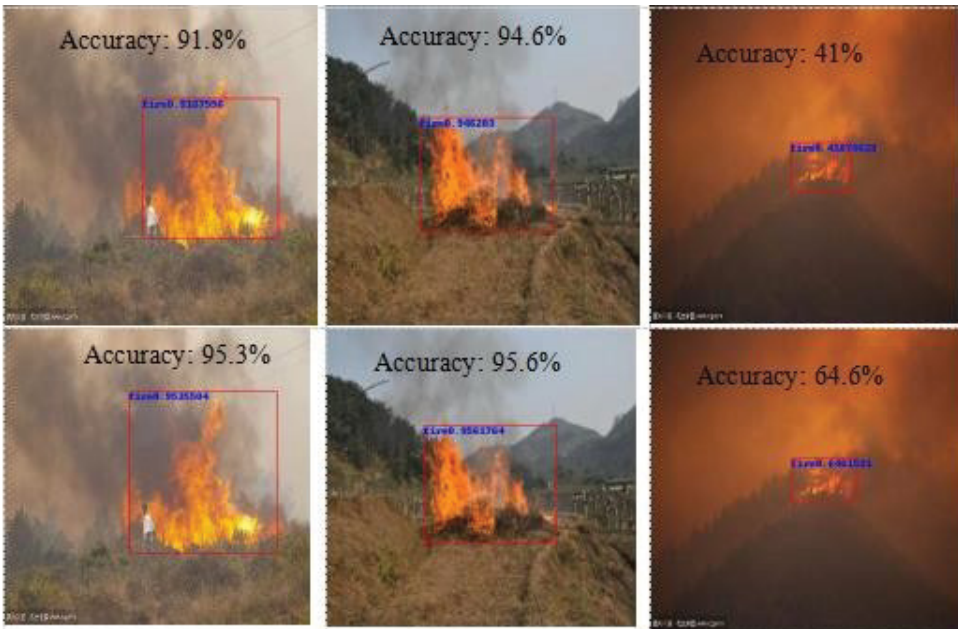

Figure 13. Pictures comparison for SSD (above) and PCA with SSD (below).

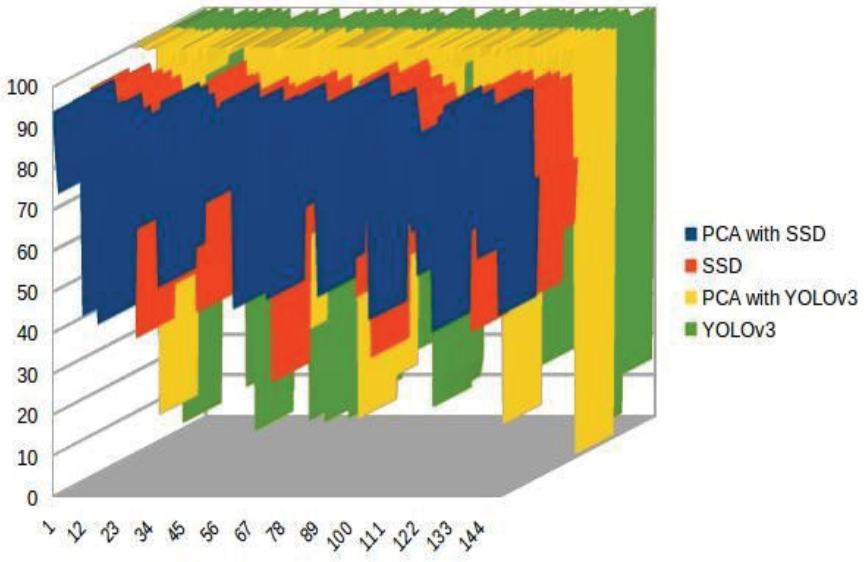

Figure 14. The detection accuracy of the four methods comparison.

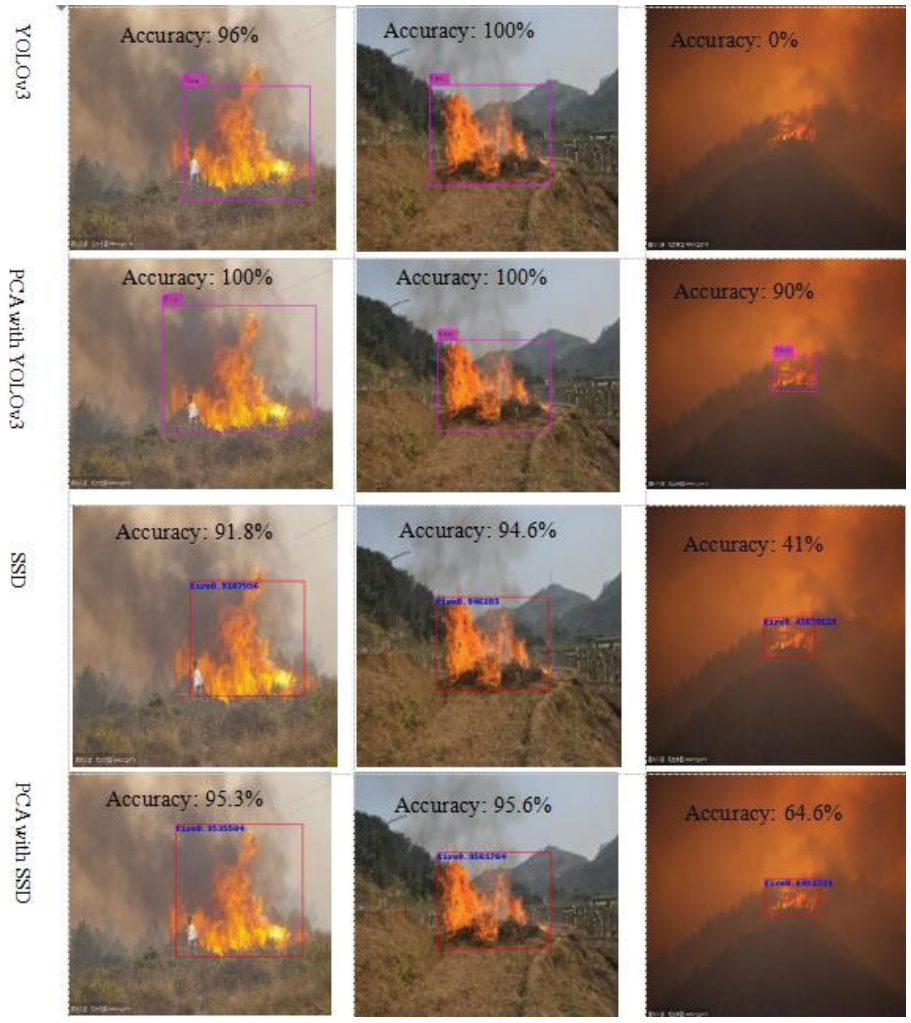

Figure 15. Four methods of image comparison. From the images comparison, you can see that the combination has better location and detection accuracy.

CONCLUSION

Previous one stage detector like YOLOv3/SSD throw the original color pictures into the deep learning network to train. In this paper, we first utilize PCA to preprocessing the original color pictures then use one stage detector to train and finally detect. The results have shown that this combination increases the mAP and detection accuracy both.

For the combination method and the individual method, we use the same configuration (learning rate, batch size, and iteration times) to implement the experiments. The motivation of this work relies on an idea that primary features in the images decide the object detection object. We use PCA to extract features from original images. For PCA with YOLOv3, we extract 260 features. For PCA with SSD, we extract 300 features. In PCA procedure. SVD is utilized to select the proper features.

The preprocessing job does not affect the final real-time detection. We only consider PCA as the processing tool, in the future, we consider that other image compression tools could also be used in the preprocessing job. We will also have tests on a different network like Faster R-CNN, Resnet-SSD, Mobilenet, resnet to implement the preprocessing job.

ACKNOWLEDGMENT

The authors would like to thank everyone, just everyone

REFERENCES

1. Kaur A, et al. Comparison of forest fire detection techniques using WSNs. Int J Comput Sci Inform Technol. 2014;5:3800-3802.
2. Ren S, et al. Faster R-CNN: Towards real-time object detection with region proposal networks. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2017;6:1137–1149.

3. Redmon J, et al. YOLO9000: better, faster, stronger. arXiv preprint. 2017.
4. Liu W, et al. SSD: Single shot multibox detector. In European conference on computer vision. 2016:21-37.
5. Lin TY, et al. Focal loss for dense object detection. IEEE transactions on pattern analysis and machine intelligence. 2018.
6. Prasad KS, et al. An autonomous forest fire detection system based on spatial data mining and fuzzy logic. Int J Comp Sci Net Secu. 2008;8:49-55.
7. Patel P, et al. Flame detection using image processing techniques. Int J Comp App. 2012;58:45–50.
8. Yadav G, et al. Optimized flame detection using image processing based techniques. Ind J Com Sci Eng. 2012;3.
9. Sam G, et al. A comparative analysis on different image processing techniques for forest fire detection. Int J Com Sci Net. 2016.
10. Anupam Mittal, et al. Forest fire detection through various machine learning techniques using mobile agent in WSN. Int Res J Eng Tech. 2016.
11. Özkana C, et al. Effectiveness of boosting algorithms in forest fire classification. The international archives of the photogrammetry, remote sensing, and spatial information sciences. 2008;37.
12. Zhang Q, et al. Deep convolutional neural networks for forest fire detection. In Proceedings of the 2016 International Forum on Management, Education and Information Technology Application. Atlantis Press 2016.
13. https://github.com/Elucidation/PCA-Image-Compression.