IJARSCT

ISSN (Online) 2581-9429

International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)

Impact Factor: 4.819

Volume 5, Issue 2, May 2021

Fire Detection using Image Processing

Shubham Chive, Aman Dubey, Ganesh Subramani, Acharna Patharne, Madhuri Kulkarni
Students, Department of EXTC
SIES Graduation School of Technology, Sri Chandrasekarendra Saraswathy Vidyapuram, Navi Mumbai

Abstract: *Fire disasters have always been a threat to homes and businesses even with the various systems in place to prevent them. They cause property damage, injuries and even death. Preparedness is vital when dealing with fires. They spread uncontrollably and are difficult to contain. To contain them it is necessary for the fire to be detected early. Image fire detection heavily relies on an algorithmic analysis of images. However, the accuracy is lower, the detection is delayed and in common detection algorithms a large number of computation, including the image features being extracted manually and using machine. Therefore, in this paper, novel image detection which will be based on the advanced object detection like CNN model of YOLO v3 is proposed. The average precision of the algorithm based on YOLO v3 reaches to 81.76% and also it has the stronger robustness of detection performance, thereby satisfying the requirements of the real-time detection.*

Keywords: Amharic, Fake News, Machine Learning, Natural Language Processing

I. Introduction

Fire alarms are present in a lot of buildings, industrial parks and workplaces. These fire alarms are usually based on sensors which detect certain characteristics of fire such as smoke, radiation, or heat. However, these fire alarms depend on the fire particles reaching the given sensor. Apart from the inherent disadvantage in the delay in detecting the fire due to the time taken for particles to reach the sensor, these alarms are basic and do not provide crucial information such as intensity, location and the size of the fire. Many of the places with a fire alarm system also have a surveillance system. These surveillance cameras can be incorporated in the fire detection process using object detection. This has become an important area of research. The object detection is based on image processing. Vision based fire detection systems have several advantages. Already installed surveillance cameras can be used for this and if they are not present, CCD (Chargedcoupled devices) cameras can be installed which are fairly inexpensive. The most important advantage is the detection time because vision-based systems do not require smoke or heat to diffuse. Another advantage is the area covered. If the camera is placed at a vantage point, it can cover a lot of open space which is a very big improvement from conventional sensors which are better in confined spaces. Lastly, in the case of a false alarm, the informed authority can check the surveillance feed to monitor the location.

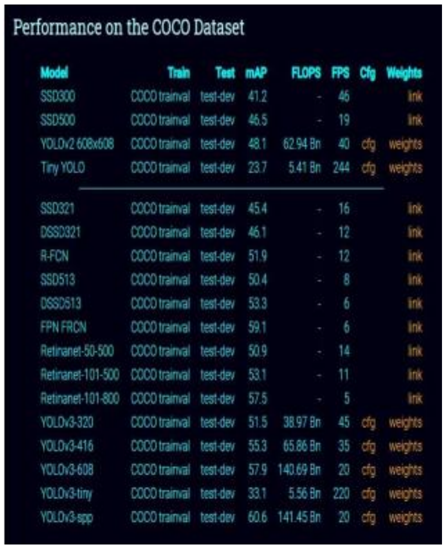
II. The Proposed Framework

2.1 Object Detection using YOLOv3

You only look once (YOLO) is an effective real time object detection system. In YOLO, a single neural network is applied to the full image. The images is divided into regions and prediction boxes with the help of network for region. Prediction probabilities weights bounding boxes.

YOLO has a lot of advantages over other systems. The predictions are based on global context in the image and it also makes predictions with a single network evaluation unlike R-CNN. This makes it a lot faster than R-CNN and even Fast R-CNN. For our project, we will be using Tiny YOLO since we will be using a raspberry pi. The Tiny-YOLO architecture is around 442% faster than other YOLO versions. The model size is small and the fast interference speed make it suitable for an embedded deep learning device such as Raspberry Pi.

The Common Objects in Context (COCO) dataset which is one of the most widely used datasets does not provide support for fire detection, so we will have to train a custom model. This can be done by creating a custom dataset. First, we collected images which fit our criteria, which is, having a fire. Then we labelled them using label.



Performance on the COCO Dataset

Model	Train	Test	mAP	FLOPS	FPS	Cfg	Weights
SSD300	COCO trainval	test-dev	41.2	-	46		link
SSD500	COCO trainval	test-dev	46.5	-	19		link
YOLOv2 608x608	COCO trainval	test-dev	48.1	62.94 Bn	40	cfg	weights
Tiny YOLO	COCO trainval	test-dev	23.7	5.41 Bn	244	cfg	weights
SSD321	COCO trainval	test-dev	45.4	-	16		link
DSSD321	COCO trainval	test-dev	46.1	-	12		link
R-FCN	COCO trainval	test-dev	51.9	-	12		link
SSD513	COCO trainval	test-dev	50.4	-	8		link
DSSD513	COCO trainval	test-dev	53.3	-	6		link
FPN FRCN	COCO trainval	test-dev	59.1	-	6		link
Retinanet-50-500	COCO trainval	test-dev	50.9	-	14		link
Retinanet-101-500	COCO trainval	test-dev	53.1	-	11		link
Retinanet-101-800	COCO trainval	test-dev	57.5	-	5		link
YOLOv3-320	COCO trainval	test-dev	51.5	38.97 Bn	45	cfg	weights
YOLOv3-416	COCO trainval	test-dev	55.3	65.86 Bn	35	cfg	weights
YOLOv3-608	COCO trainval	test-dev	57.9	140.69 Bn	20	cfg	weights
YOLOv3-tiny	COCO trainval	test-dev	33.1	5.56 Bn	220	cfg	weights
YOLOv3-spp	COCO trainval	test-dev	60.6	141.45 Bn	20	cfg	weights

Figure 1: Graphical image annotation tool and it helps to label object bounding boxes in images.

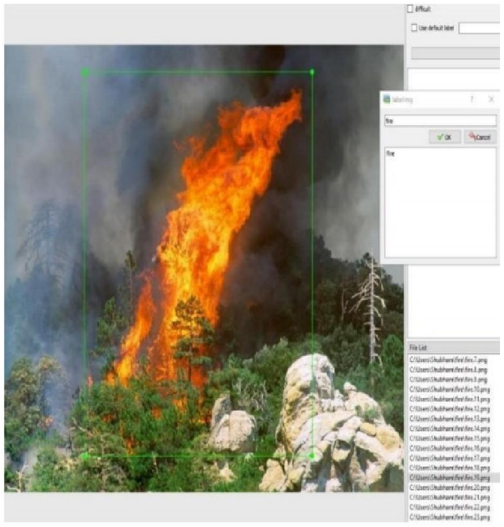
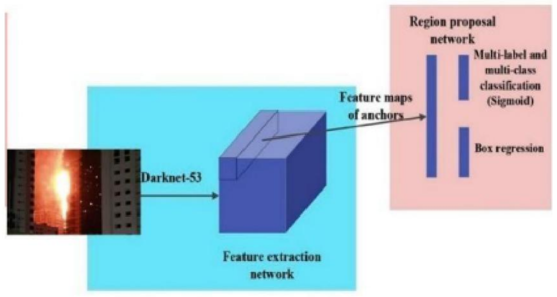


Figure 2: GUI used to created bounding boxes in images

III. YOLO v3 Algorithm

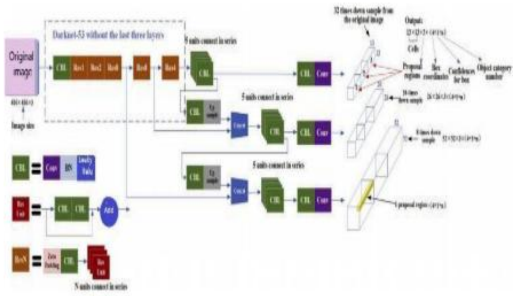

Region proposal network
Multi-label and multi-class classification (Sigmoid)
Feature maps of anchors
Box regression
Darknet-53
Feature extraction network

YOLO v3

Diagram of the fire algorithm based on yolo v3

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Impact Factor: 4.819
Volume 5, Issue 2, May 2021

To generate small scale feature map, Darknet-53 is being used by YOLO v3, which is from the original it is 32 times down sampled. For example, the size of the feature map is 13×13 if the size of the original image is 416×416. The small-scale feature map is used to detect large objects. By up sampling the small-scale feature map and concatenating with a feature map from an earlier layer a large-scale feature map is generated by YOLO v3. Small objects are detected by using complex features of deeper layer and location information of the earlier layer from the large scale feature map. From the original image the three scales of feature maps are 8, 16, and 32 times down sampled. There are N units of Res Unit connected in series in ResN. Concentration operation is denoted by concat. This concat is different from the Add operation in residual layers. Dimensions of the feature maps are expanded by the feature maps. On the other hand, Add operation is just adding the dimension without changing them. To predict the multilabel classification per bounding box, independent sigmoid function is used by YOLO v3. This means that per bounding box could belong to multiple categories like fire and smoke. Regions where fire and smoke appear simultaneously, this design is useful for detecting them.

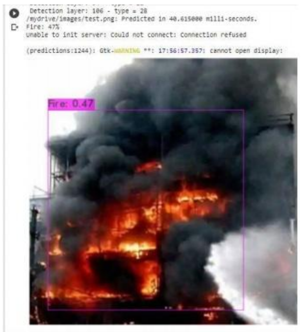

Original image
Darknet-53 without the last three layers
CBL
Add
Concat
N units connect in series

- **CBL:** The smallest component in the Yolov3 network structure, by Conv +BN +Leaky relu The activation function consists of three.
- **Res unit:** Learn from Resnet The residual structure in the network allows the network to be built deeper.
- **ResN:** By one CBL with XA residual component constitutes a large component in Yolov3. The CBL in front of each Res module plays the role of downsampling, so after 3 times of Res module, the resulting feature map is 416 > 52 > 26 >13

IV. Training Algorithm

Fire Image Dataset

A large number of data is required for fire images dataset for the training of algorithms witch are based on CNNs. However, current small scale images/video fire databases cannot meet the needs. Table 1 shows some small scale dataset for images/videos. Therefore in this paper we collected and labelled 1400 such images to give a good foundation for our dataset. For our convenience, to test the code and to create a custom model, we used Google Colab. Our custom object detector was trained using this dataset using Darknet. The following output is one of the examples observed.



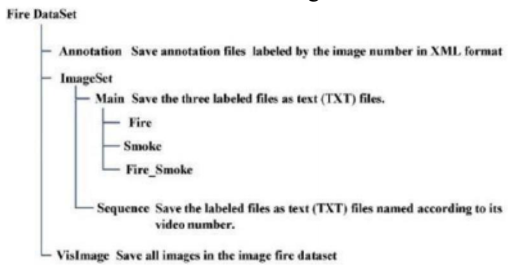
```
Detection layer: 106 - type = 28
/mydrive/images/test.png: Predicted in 40.615000 milli-seconds.
Fire: 47%
unable to init server: Could not connect: Connection refused

(predictions:1244): Gtk-WARNING **: 17:56:57.357: cannot open display:
```

Object detection output from Google Colab

Institutions	Format	Object	Website
Kaggle	Image	fire, smoke, disturbance	https://www.kaggle.com/phylake1337/firedataset
National fire research laboratory, NIST	Image	fire	https://www.nist.gov/topics/fire
State Key Laboratory of Fire Science, University of Science and Technology	Image	Fire, smoke	https://smoke.ustc.edu.cn/datasets.htm

Table 1: Small scale fire image/video databases


Fire DataSet
Annotation Save annotation files labeled by the image number in XML format
ImageSet
Main Save the three labeled files as text (TXT) files.
Fire
Smoke
Fire_Smoke
Sequence Save the labeled files as text (TXT) files named according to its video number.
VisImage Save all images in the image fire dataset

Structure of the fire image dataset.

Scenario	Objects		Disturbances		Images
	Smoke	Fire	Smoke-like	Fire-like	
Indoor	376	576	541	539	634
Outdoor	295	793	235	696	766
Total	671	1359	776	1235	1400

Table 2: Small scale fire image/video databases

V. Results And Discussion

5.1. Performance of Testset1

Testset1 is a benchmark fire image database consisting of 700 images. This database has a 578 fire images and 122 images containing no fire. The number of videos played 8, Number of true detection of fire in videos 59, Number of false detection 19, Number of true false 5, The Percentage of true detection 71.08%.

In this testset1 the number of true detection is less than the expected true detection, Therefore, a more detailed evaluation is conducted in the testset2

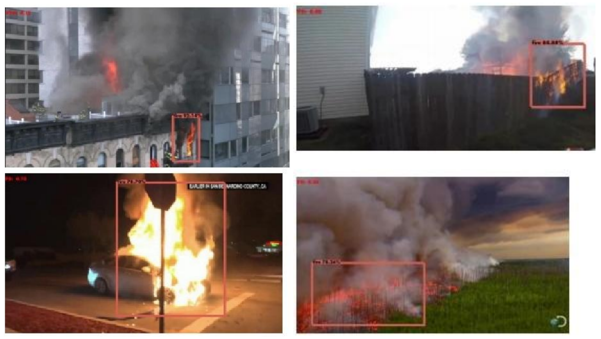


Figure 9: Output of Testset1

We tested our trained model on video of fire, the image shows that the fire is detected by a red rectangle with the percentage of the fire that has been detected, the output of testset1 shows that the fire is detected but it fails to detect the fire shadowed by the smoke. So it is important that the smoke covering fire should be detected.

5.2 Performance of Testset2

Testset2 is a benchmark fire image database consisting of 1400 images, which includes 478 smoke samples and 896 fire samples. Testset2 is very challenging as it collects images from more scenarios containing a large number of smoke-like and fire-like disturbances. Therefore, it is more suitable for evaluating the performance of the proposed algorithms.

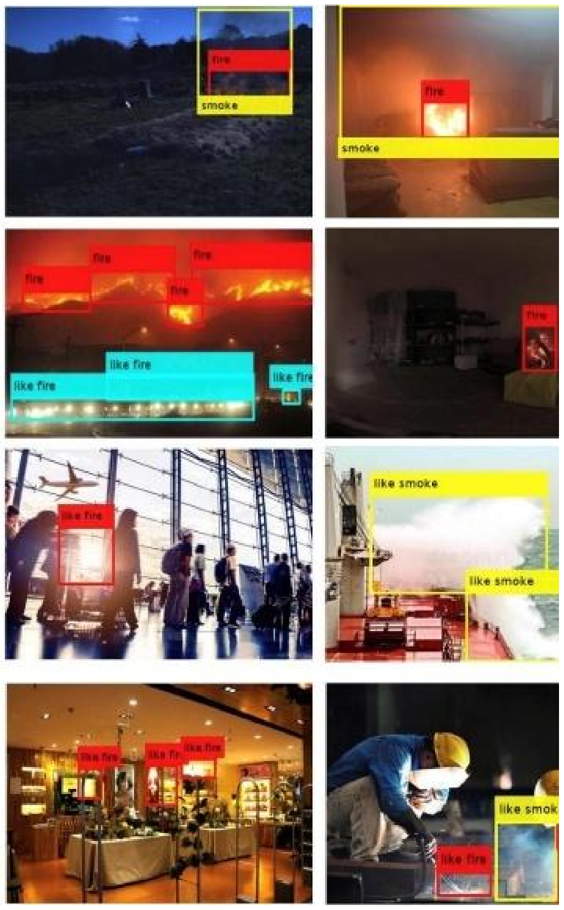


Figure 10: Output of Testset 2

The previous output fails to detect the fire overshadowed by smoke, So we trained our model by adding the smoke dataset to the fire dataset. The output of Testset 2 shows that the fire and smoke is detected.

Quantitative Analysis of Results

A common metric to measure the object detection algorithm is intersection over union (IOU). IOU is a metric that finds the difference between ground truth annotations and predicted bounding boxes. In object detection, the model predicts multiple bounding boxes for each object, and based on the confidence scores of each bounding box it removes unnecessary boxes based on its threshold value.

IOU=Area of union/area of intersection

F1= Weighted average of precision and recall

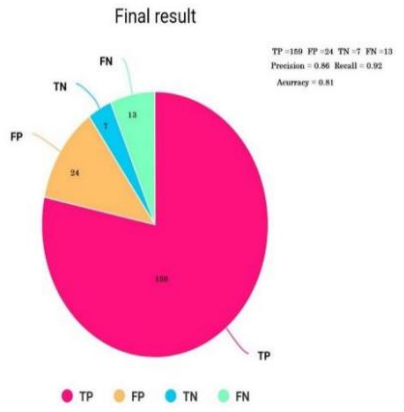
$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

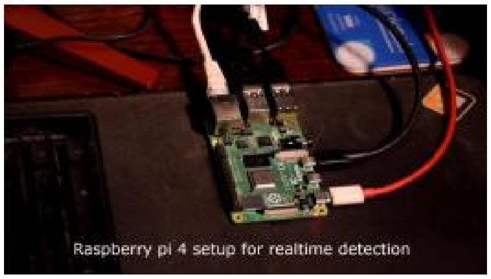
- *TP* = True positive
- *TN* = True negative
- *FP* = False positive
- *FN* = False negative

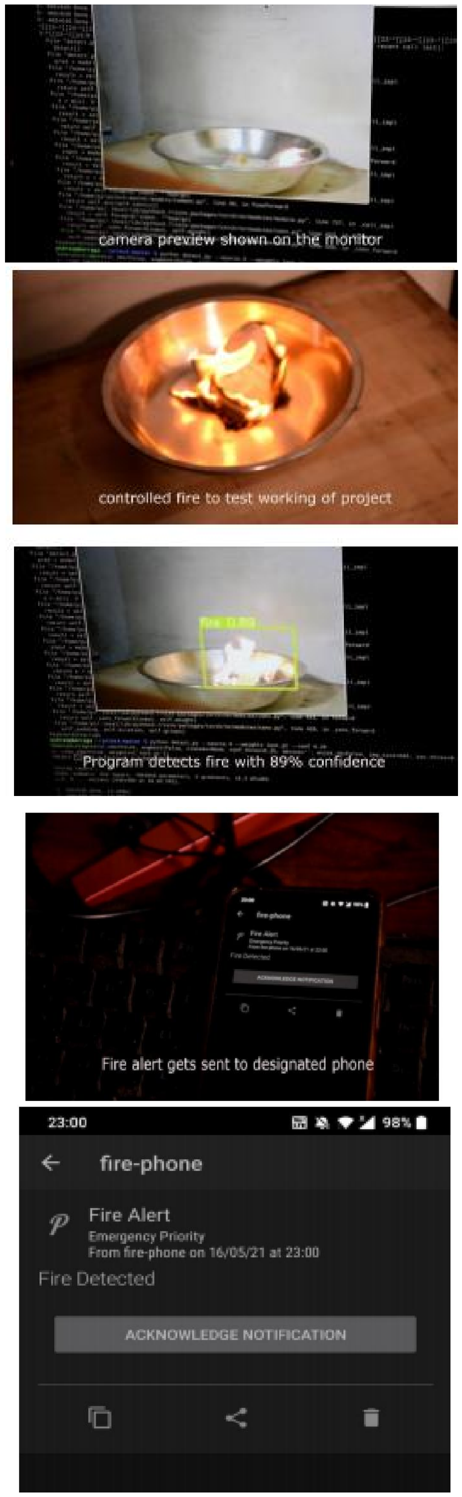
True Positives (TP) - These are the correctly predicted positive values which means that the value of actual class is yes and the value of predicted class is also yes.
True Negatives (TN) - These are the correctly predicted negative values which means that the value of actual class is no and value of predicted class is also no.
(False positives and false negatives, these values occur when your actual class contradicts with the predicted class.)
False Positives (FP) – When actual class is no and predicted class is yes.
False Negatives (FN) – When actual class is yes but predicted class in no
Precision: Precision measures how accurate your predictions are. i.e. the percentage of your predictions are correct. It measures how many of the predictions that your model made were correct.
Recall: Recall is the ratio of correctly predicted positive observations to the all observations in actual class



The total number of times, these images were played was 203 times. Out of those 203 times, true positive value was 159, value of false negative was 13. The value of false positive and true negative were 24 and 7 respectively. The accuracy came out to be 81%

5.3. Real-time Output




camera preview shown on the monitor
controlled fire to test working of project
Program detects fire with 89% confidence
Fire alert gets sent to designated phone
23:00
98%
ACKNOWLEDGE NOTIFICATION

fire-phone

Fire Alert
Emergency Priority
From fire-phone on 16/05/21 at 23:00
Fire Detected

Figure 11: Output of our project in real time

The Raspberry pi is connected to the internet, Program is running on the Pi, A image preview from the camera which will be used to detect fire. Camera detects frame every 3 sec and as soon as fire is detected a rectangle with the percentage of fire detected appears then the program will send a notification to the phone through an app called pushover api the notification send will be an emergency priority, so it will not go away till the user acknowledge the notification.

VI. Conclusion

In chapter one we mentioned the disadvantages of the existing fire detector. We mentioned how the existing fire detector had difficulties detecting the fire which are out of its range. Also, we mentioned the objectives and scope of our projects. In chapter two we did the literature survey of based on previous different research paper. In the third chapter we explained how this project will be CNN based to detect fire. Also, we explained the object detection using yolov3. We also used it on some image samples and showed the output. With the help of block diagram and flowchart we explained the working of the system. In the fourth chapter we explained what type of camera we will be using for images. Also, we explained the use of GPS and how we will use tiny yolo for implementation on raspberry pi. Advanced object detection CNNs YOLO v3 is used to improve the performance of image fire detection technology to develop algorithms of image fire detection. Complex image fire features and detected fire in different scenes can automatically be extracted with the help of proposed algorithms. The evaluation experiments results are given as follows:

1. In this testset1 the number of true detection is less than the expected true detection
2. The highest accurate algorithm based on YOLO v3, with 81.7% accuracy, detects fire the most quickly and is the strongest robust
3. True detection of testset2 has the highest accuracy than the previous testset1

References

[1] ByoungChul Ko, Sooyeong Kwak, "Survey of computer vision-based natural disaster warning systems," Opt. Eng. 51(7) 070901 (28 June 2012)

[2] Thou-Ho Chen, Ping-Hsueh Wu and Yung-Chuen Chiou, "An early fire-detection method based on image processing," 2004 International Conference on Image Processing, 2004. ICIP '04., Singapore, 2004, pp. 1707-1710 Vol. 3, doi: 10.1109/ICIP.2004.1421401.

[3] Ko, Byoungchul & Cheong, Kwang-Ho & Nam, Jae-Yeal. (2010). Early fire detection algorithm based on irregular patterns of flames and hierarchical Bayesian Networks. Fire Safety Journal -FIRE SAFETY J. 45. 10.1016/j.firesaf.2010.04.001.

[4] Qi, X. & Ebert, Jessica. (2009). A computer vision-based method for fire detection in color videos. International Journal of Imaging. 2. 22-34.

[5] Zhang J., Zhuang J., Du H., Wang S., Li X. (2006) A Flame Detection Algorithm Based on Video Multi-feature Fusion. In: Jiao L., Wang L., Gao X., Liu J., Wu F. (eds) Advances in Natural Computation. ICNC 2006. Lecture Notes in Computer Science, vol 4222. Springer, Berlin, Heidelberg.

[6] Z. Yin, B. Wan, F. Yuan, X. Xia and J. Shi, "A Deep Normalization and Convolutional Neural Network for Image Smoke Detection," in IEEE Access, vol. 5, pp. 18429-18438, 2017, doi: 10.1109/ACCESS.2017 .2747399.

[7] K. Muhammad, J. Ahmad, I. Mehmood, S. Rho and S. W. Baik, "Convolutional Neural Networks Based Fire Detection in Surveillance Videos," in IEEE Access, vol. 6, pp. 18174-18183, 2018, doi: 10.1109/ACCESS.2018.2812835.

[8] Li, Pu & Zhao, Wangda. (2020). Image fire detection algorithms based on convolutional neural networks. Case Studies in Thermal Engineering. 19. 100625. 10.1016/j.csite.2020.100625.