TEST Engineering & Management

May - June 2020
ISSN: 0193-4120 Page No. 5234-5238

YOLO Based Real time Forest Fire Detection

Shireesh Kumar P A[1], Keerthi Reddy[2], Surekha Thota[3], Shreya Merin[4], Thejas HR[5]

[1,2,3,4,5]School of Computing and Information Technology
REVA University, Bangalore, India
[1]shireeshkumar.1998@gmail.com, [2]Keerthidfg.999@gmail.com, [3]surekha.thota@reva.edu.in,
[4]shreyamerin1@gmail.com, [5]Thejas.hr07@gmail.com

Article Info
Volume 83
Page Number: 5234-5238
Publication Issue:
May - June 2020

Article History
Article Received: *19November 2019*
Revised: *27 January2020*
Accepted: *24 February 2020*
Publication: *16 May 2020*

Abstract

Forest fires are very hazardous and cannot be ignored knowing the fact that they have the potential to tear down the balance between flora and fauna in the distant future. In this paper the discussion related to a popular model based on AI i.e. YOLO (You Only Look Once), used for the detection of the fire outbreaks those which mainly involve Wildfires is put together. To generalize, the power of AI is incorporated to detect and help in taking early precautions from the immense damage that could be caused by the wildfire outbreaks. The paper begins with introduction initially comprising with history of fire outbreak incidents, causes and its after effects. The literature survey references lead to deep understanding of previous models and its limitations which are key points for further enhancements. Our outline work has been explained in phases: dataset pre-processing, environment setup and YOLOv3 model training and validation. By exploiting YOLOv3 this paper proposes an algorithm which is capable of detecting real time scenario which in our case is forest fire with required precision.

Keywords: *YOLOv3, Real-time detection, Deep Learning, Wildfire*

1. Introduction

Fire breakouts cause major damage to the environment and also to the site where they occur. One of the biggest consequences of fire is known to be forest fire outrage, which causes a huge loss to the environment and destruction to large ecosystems. Statistics have shown that forest fires have increased from 50,477 to 58,083 from the years 2018 to 2019 respectively, according to the National Interagency Fire Center (NIFC)[1]. An average of 7 million acres of forest land burns down every year by 72,400 fires since the year 2000[2]. Reports state that over 18 million hectares have burned in the Australian bush fires, destroying over 5,900 buildings and 2800 homes. Millions of animals have lost their habitats whereas a few more are on the verge of extinction[3].The Amazon rainforest, known for its highest biodiversity in the world, plays a great role in fighting againstclimate change by absorbing carbon dioxide[4].According to the space research agency INPE, the number of forest fires in the amazon has increased by 30.5% from 2018 to 2019[5].Once a fire has occurred, it can burn for hours resulting in unrecoverable damage before it is detected by the

appropriate authorities and actions are taken to put them out. Forest fires take a significant amount of time to be noticed owing to it's remote location and rough terrain. Control of fire spread in the ignition phase is easier than in the phase where fire becomes uncontrollable and the damage is excessive. In other words, damage increases exponentially with time.

According to paper[5238] ,there are 4 existing technologies deployed for fire detection today. They are as follows :

i)Human Observation : The fire or smoke is observed by a human and reported to the fire department. Although this is a reliable method, it is very inefficient. Exact location is hard to trace and the delay in detection results in damage that has already taken place.

ii)Satellite Systems : Fire is detected via the satellite images taken by special satellites set up for the purpose of detection. There are several drawbacks to this system. It is difficult for the satellite to cover the entire region or even

May - June 2020
ISSN: 0193-4120 Page No. 5234-5238

provide continuous coverage. The detection is not always real time and images are mostly disrupted by bad weather conditions.

iii)Optical cameras : The camera equipped with geographical maps, captures images once in a while and finds the pixels containing smoke or fire. But these cameras raise many false alarms due to shadows, wind, etc. This technology is limited to line of sight observance and therefore trees of varying heights cannot be observed. These cameras also require manual instalment in thousands of kms of forest spread which means that the number of camera units required also increases.

iv)Wireless sensor networks : These are a network of sensors with a camera that is linked through a protocol.

The sensors detect fires and alerts the camera in the vicinity to turn on, providing real time images of the fire. The low chance of fire alarms and high efficiency and localization, this system is one of the best systems available for fire detection. But errors can occur in data compilation and it also raises energy conservation problems.

Keeping all the limitations of the current systems of forest fire detection in mind, this project aims to come up with a solution that is quick, efficient and economical by detecting the fire in its onset, thereby bringing it to the notice of the fire department to take necessary actions before it's too late. This project shows a promising future in detecting other kinds of fire outbreaks in various other locations in real time.

In this paper, studies made relatedto YOLO technology will be discussed. Sucessively, the methodology will consist of the following chapters:

i)Dataset and Pre-processing

ii)Environment setup

iii)Training the model

The model is tested and the results obtained is graphed over which the accuracy is calculated.

2. Literature Survey

This field discusses the quality efforts made by the data scientists, field workers in finding the various methodologies that could help in the monitoring, detecting of the wildfire outbreaks, and help in taking early precautions from the immense damage that could be caused from this. Chen, Y., Zhang paper [7] discussesthe thermal image processing and usage of CNN(Convolutional Neural Network)deep learning model to detect the fire outbreaks in an image that are been captured through camera's fitted onto a UAV. Similar work considering CNN is also discussed in paper [8][9][10] where the CNN model has been trained over the data related to the past outbreaks of wildfire and have discussed how early detection can be found helpful in taking preliminary precautions to avoid the immense damage caused. But one major drawback that is been mentioned in these papersis that when it comes to real-time detection, they are not found to be enough accurate or stable and were computationally too expensive.

In order to overcome the accuracy issues in detecting any such objects higher version of CNN were developed [11] like RCNN, Fast CNN, SPP-net. RCNN uses an approach called selective search method that sets the bounding boxes over an image to 2000 in order to improve the accuracy and the image with the bounding box on it is fed onto the connected neural network for detection, but this grid structure of 2000 over an image was too large for the neural network to process it so further SPP-net was used that was similar to R-CNN but used spatial pooling over the last layer of the neural network over the traditional max pooling where pooling is gathering only the pixel values that detect required feature but this was no trivial to perform backpropagation which is updating the weights of the neural network in backward direction which is found very necessary in getting desired accuracy without having to perform redundant training just to update the training values.Even though other technologies were developed but the main drawback faced in them were too much pre-processing of the input data was required in order to detect the required object and also the fact that this models like RCNN, SPP-net and Fast CNN required the construction of several layers of neural networks with a well defined number of nodes in each layer in order to get the required accuracy and also this model were to be designed differently when they are made to work on colored images and grey scaled images, which was found to be an overhead.

The Latest development in this field as mentioned in paper [12][13] discusses the evolution of the models like YOLO v1, v2 and the latest v3 help us understand the improvements with respect to the above drawbacks and how these models help in accurate detection with less pre-processing of the input data and also how it's well defined 53 layers of neural network layers help increase the processing speed and precise detection of the objects suiting the application requirements. Moreover, the single shot detection features of the YOLO v3 the latest version which is one of its key feature makes an accurate detection of objects in a single forward pass in the neural network without excess processing and is found very much helpful in the real time analysis.Also that YOLO model is capable enough to handle both colored and grey scaled images with out any excess computation,this feature motivated us to use this technology in the real-time detection of the outbreaks of wildfires and also made capable of detecting even the fire outbreaks in residential fortresses, college buildings, office building etc. Our work mainly focusses on how this technology could be utilized both on a large scale and small scale areas in real time.

3. Methodology

A. Dataset Pre-processing

Data pre-processing involves the following steps as mentioned in the

Figure 1.Initially as we begin with the data collection we gather the required data images for our purpose of detecting the forest fire outbreaks. The data gathered here

TEST Engineering & Management

May - June 2020
ISSN: 0193-4120 Page No. 5234-5238

is manually collected and does not refer to any standard dataset. The collection not only comprises imagesof wildfire outbreaks but have also the images of the fire outbreaks in residential areas, colleges or office buildings etc. in order to make this application useful in both large scale and small-scale detection.

Once the data is collected, we label the data with the required class i.e. labelling the image as fire if fire outbreak is present in it and if not present it is left without any changes so that we make it easy for the model to focus on only the required class and leave the rest unclassified. This way the model does not waste time in classifying the useless data. For labelling the data we use the software called LABELIMG.The label for each image is stored as annotations in xml format and loaded onto the model together with their corresponding image.

Once the data is labelled, we split them for training and validation purpose then load them both onto YOLO model so that it trains over the training data and tests over the validation set which generates the loss entropy for the analysis of accuracy.This loss entropy is discussed further in the methodology. The total size of the dataset is 525 in which 325 are used for the training purpose and remaining for validation set.

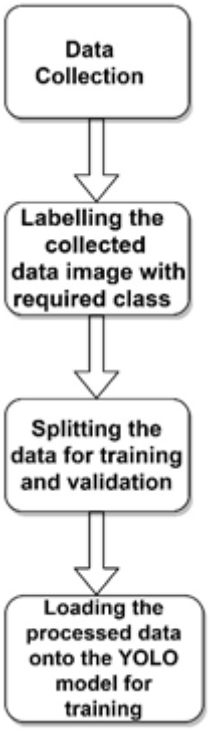

Figure 1:Data Preprocessing

B. Environment Setup

In this section we discuss the required platform for the training of the YOLO model, and the supporting modules and libraries used for the training purpose. To begin with, Google Colab cloud environment is used which provides GPU facility that increases the pace of training the model, and to make use of the GPU support TensorFlow-GPU V1.13.1 is installed on to this Colab platform. Also, to be mentioned that Colab is a python supported platform so the code is written in python language.

The library called IMAGEAI is acquired to construct the YOLO model which is embedded with the open CV technology which is used in real time detection of objects along with the trained model. The Special feature of IMAGEAI is that it can detect multiple occurrences of the required object in a single frame of the image easily.

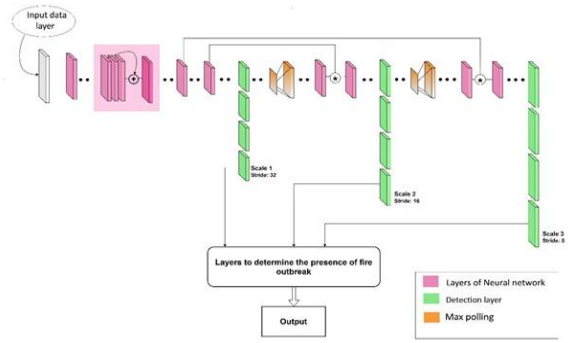

Figure 2:YOLO v3 Architecture

C. Training the Model

Once the above environment is setup, we are ready for construction of the YOLO model and training it over the data-set pre-processed earlier.

Before we get into the analysis of the code used for the training, it is important to understand the background working of the YOLO model and its architecture in **Error! eference source not found.**.

Firstly IOU (Intersection Over Union) computation needs to be understood. IOU is the difference between ground truth box and detected truth box i.e. when the data is labelled, we mark the part in the image where the fire is detected and it is called as ground truth box and the area that the model detects consisting of fire outbreak is known to be detected truth box. The difference in the area and the position of their areas with respect to its coordinates are together computed and known to be IOU. The IOU takes values ranging between 0 to 1 and the more it is close to 1 the more accurate will the model be. **Error! Reference source not found.** represents the eneral representation of IOU.

$$IOU = \frac{\text{Area of Intersection}}{\text{Area of Union}}$$

Figure 3:

The YOLOv3 model constructed consists of 53 layers of neural network, which is further grouped under 3 subgroups. Each of the subgroup acts as the detection layer with different stride value 32, 16, 8 as showed in **Error! Reference source not found.**. This stride value is he size of the feature matrix used to capture only the pixel

TEST Engineering & Management

May - June 2020
ISSN: 0193-4120 Page No. 5234-5238

corresponding to required object which has to be detected. The stride value is decreased towards the end in order to help model detect the smaller objects also in the image. Max pooling which is another term used here is one among the mathematical computation needed to increase the accurate rounded values generated during training.

It is appropriate to discuss the other important computation related to the loss entropy and the accuracy analysis after analysing the training code.

```
From imageai.Detection.Custom
 importDetectionModelTrainer

trainer = DetectionModelTrainer

(trainer.setModelTypeAsYOLOv3()
```

Figure 4: Initialising the YOLO model

The above **Error! Reference source not found.** code hich is part of paper implementation refers to defining the YOLO model and make it ready for training. Trainer variable is the variable that is assigned to the YOLO model configuration which helps us train the YOLO model over the dataset.

```
trainer.setDataDirectory(
data_directory="fire-dataset")
trainer.setTrainConfig(
object_names_array=["fire"],
batch_size=4, num_experiments=200,
train_from_pretrained_model=
"pretrained-yolov3.h5")
```

Figure 5: Loading the dataset

The above code in **Error! Reference source not ound.** refers to loading of data set onto the YOLO model, here the fire-dataset is the name has been uploaded and only one class name is defined called “fire” which is used to classify the images in the dataset and also the number of iterations on count 200 and also the batch-size of 4.Batch size indicates the number of images to be considered for each iteration.

Once this code is executed, it’s time to train the model using this line of code: **trainer.trainModel().**

After each iteration of the training, a batch of images from the validation set are taken and the loss value of 3 set of layers is computed and displayed. Once the training is completed, we take 3 sets of the neural network layer’s loss values of 200 iterations and the training loss graph is plotted in **Error! Reference source not found..**

Figure 6: Loss Entropy

The loss is computed by the YOLO itself which makes it easy for the trainer avoiding the manual work. The loss function that is used for this purpose is the SUM SQUARED ERROR (SEE). This function is an easy to optimize mathematical code so it is preferred here in YOLOv3.Here the lesser the loss value the more accurate the model is found.

Loss function here is the sum of three parts which are the localization loss, confidence loss and classification loss **Error! Reference source not found.**.The first term n this **Error! Reference source not found.** denotes the localization error which deals with the error in classification in terms of the differences with respect to the ground truth box. Second term here is similar to the first term only but they deal with only the differences in the coordinates of the ground and detection truth box.

Term 3 and 4 in **Error! Reference source not ound.**represents the confidence loss and further the term 5 refers to the classification loss. This loss computations requires a high level of mathematical knowledge to understand but since we are more focused on the implementation, we have just mentioned its mathematical background in terms of formula.

All of three categories of losses is combined to get the final loss entropy value and the lesser this value the more accurate the model turns out in the end. Further the loss entropy and the accuracy are discussed in the results section.

4. Results and Discussion

After training all the three set of loss values for every iteration, it is plotted in the graphical form and also same process is followed for the validation set of 200 images and testing loss value graph is computed **Error! eference source not found.**. We combined all the three categories of loss of each iteration into an average loss and again average loss of all the iterations is combined into a final average value and is found to be 1.250 and when similar procedure is followed for the testing, we found the loss entropy to be 1.440. Since the loss entropy of both training and testing is considerably very small, we consider our model to be trained appropriately and

May - June 2020
ISSN: 0193-4120 Page No. 5234-5238

accurately as loss value is inversely proportional to accuracy.

The loss entropy in both the training and testing phase are efficient enough for our 525 images dataset but we cannot conclude since our dataset size is not large enough and it is recommend to use large size dataset to cover all kinds of scenarios of fire outbreaks and this will eventually help in the accuracy during the real time.

Figure 7: Final output

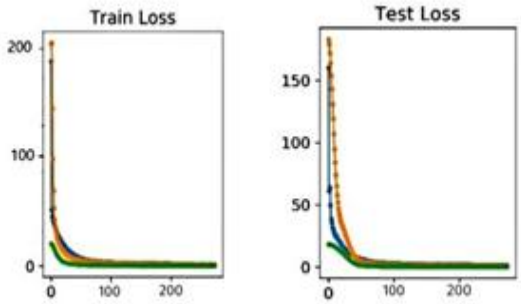

Figure 8: Training and Validation

5. Conclusion and Future Outlook

YOLO model is more sophisticated than we assume it to be when it comes to implementation and requires a larger number of data to get well trained in order for it to sustain the accuracy over the real time and for this we need a higher computational power of GPU which will helpus deal with the processing of high resolution images also.

In this paper we tried to present a simple implementation of YOLO in-order to analyze it's efficiency over the object detection. As for the YOLO it has a promising future in the field of object detection but only requires higher computational power in order to make it more efficient for real time.

References

[1] https://www.iii.org/fact-statistic/facts-statisticswildfires
[2] https://fas.org/sgp/crs/misc/IF10244.pdf
[3] https://www.unenvironment.org/news-and-stories/story/ten-impacts-australian-bushfires
[4] https://towardsdatascience.com/an-analysis-of-amazonian-forest-fires-8facca63ba699
[5] https://www.ndtv.com/world-news/amazon-forest-fires-increased-by-30-in-2019-report-2161048
[6] https://journals.sagepub.com/doi/full/10.1155/2014/597368
[7] Chen, Y., Zhang, Y., Xin, J., Wang, G., Mu, L., Yi, Y., ... & Liu, D. (2019, June). UAV Image-based Forest Fire Detection Approach Using Convolutional Neural Network. In 2019 14th IEEE Conference on Industrial Electronics and Applications (ICIEA) (pp. 2118-2123). IEEE.
[8] Kinaneva, D., Hristov, G., Raychev, J., & Zahariev,P. (2019,May). Early Forest Fire Detection Using Drones and Artificial Intelligence. In 2019 42nd International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO) (pp. 1060-1065). IEEE.
[9] Muhammad, K., Ahmad, J., Lv, Z., Bellavista, P., Yang, P., & Baik, S. W. (2018). Efficient deep CNN-based fire detection and localization in video surveillance applications. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 49(7), 1419-1434.
[10] Shafiee, M. J., Chywl, B., Li, F., & Wong, A. (2017). Fast YOLO: A fast you only look once system for real- time embedded object detection in video. arXiv preprint arXiv:1709.05943.
[11] Du, J. (2018, April). Understanding of Object Detection Based on CNN Family and YOLO. In Journal of Physics: Conference Series (Vol. 1004, No. 1, p. 012029). IOP Publishing.
[12] Jiao, Z., Zhang, Y., Xin, J., Mu, L., Yi, Y., Liu, H., & Liu, D. (2019, July). A Deep Learning Based Forest Fire Detection Approach Using UAV and YOLOv3. In 2019 1st International Conference on Industrial Artificial Intelligence (IAI) (pp. 1-5). IEEE.
[13] Ma, S., Zhang, Y., Xin, J., Yi, Y., Liu, D., & Liu, H. (2018, June). An early forest fire detection method based on unmanned aerial vehicle vision. In 2018 Chinese Control And Decision Conference (CCDC) (pp. 6344- 6349). IEEE.
[14] Zhao, Z. Q., Zheng, P., Xu, S. T., & Wu, X. (2019). Object detection with deep learning: A review. IEEE transactions on neural networks and learning systems, 30(11), 3212-3232.
[15] Womg, A., Shafiee, M. J., Li, F., & Chwyl, B. (2018, May). Tiny SSD: A tiny single-shot detection deep convolutional neural network for real-time embedded object detection. In 2018 15th Conference on Computer and Robot Vision (CRV) (pp. 95-101). IEEE.
[16] Pal, S., & Chawan, P. M. (2019). Real-time object Detection using Deep Learning: A survey.
[17] Sarkale, A., Shah, K., Chaudhary, A., & PN, T. (2018). A Literature Survey: Neural Networks for object detection. VIVA-Tech International Journal for Research and Innovation, 1(1), 1-9.
[18] Chahal, K. S., & Dey, K. (2018). A survey of modern object detection literature using deep learning. arXiv preprint arXiv:1808.07256.