AN ROBUST AND DYNAMIC FIRE DETECTION METHOD USING CONVOLUTIONAL NEURAL NETWORK

SUGANIA S.S [1], NAVEEN KUMAR .S[2], PRANAV J.S[3]

[1]Assistant Professor, Department of CSE, Jeppiaar SRR Engineering College, Padur, Chennai.
[2, 3]Final Year Student, Department of CSE, Jeppiaar SRR Engineering College, Padur, Chennai.

---***---

Abstract - *Vision based fire detection framework has lately picked up popularity when contrasted with customary fire recognition framework dependent on sensors. The need of video perception at private, Modern, business regions and woods areas has expanded the use of vision-based fire acknowledgment system. Recently lots of fire related accidents has occurred due to improper Surveillance or unable to cover those uncertain regions like restricted areas in forest or any factory buildings. In order to overcome such accidents , we propose a new method using Convolutional neural networks (CNN). The critical issue with CNN-based fire disclosure structures is their execution progressively frameworks, in view of their high memory and computational necessities for usage. Right now, we propose a unique, well disposed, and computationally effective CNN design, using You only look once (YOLOv3) at an image to predict what objects are present using a single convolution network. which reduces the computational time and cost and improves the accuracy and reduces false alarm.*

Key Words: Convolutional neural networks, Yolov3, Fire detection, Fire Accidents, Machine learning.

1. INTRODUCTION

A convolutional neural is a machine learning algorithm used for image classification and it has several layers for processing an image, just as numerous shrouded layers. The shrouded layers of a CNN regularly comprise of a progression of convolutional layers that convolve with a duplication or other speck item. The initiation work is ordinarily a RELU layer, and is in this way followed by extra convolutions, for example, pooling layers, completely associated layers and standardization layers, alluded to as shrouded layers on the grounds that their sources of info and yields are veiled by the enactment capacity and last convolution. Accordance to the recent times, lots of fire disaster and fire related accidents have occurred due to lack of surveillance in the unpredictable regions these kind of fire accidents occurs, As indicated by the ongoing insights Australia has become an inferno. The greater part of the nation is gagging on smoke, and the skies shine orange as bushfires keep on assaulting the landmass. Not just Australia, Amazon Rain Forest likewise consumed to cinders, On December 8, 2019, upwards of 43 individuals were died in a frightful fire mishap in Delhi. There was an incident apparently begun in a 4-story unlawful production line unit in Anaj Mandi, Delhi. In that incident almost 50 workers were encountered with dozing in the confined space. Although it can be totally avoided if the fire is detected at the starting stage, So we propose a new model or method to avoid such disasters without the need of human surveillance.

2. AN INTRODUCTION TO A NEW CONCEPT

You Only Look Once (YOLOv3) is an object detection algorithm (based on neural nets) which can be used detect objects in live videos or static images, it is one of the fastest and accurate object detection method up to date. YOLO utilizes a preparation set included pictures and their relating bouncing boxes. YOLO runs at its best while using a NVIDIA GPU, on the off chance that you run YOLO on CPU alone, you will encounter fundamentally more slow preparing occasions and investigating times. This approach involves a single deep convolutional neural network that splits the input into a grid of cells and each cell directly predicts a bounding box and object classification. Due to the smoke and flame are not fixed in shape, they could appear in various sizes. Therefore, two YOLO layers are used for the detection to speed up the algorithm. Besides, the second feature extraction is performed on layer 27, resulting in an output of 26�6�4. 24 represents the network tensor size, which can be calculated.

3. SYSTEM IMPLEMENTATION

The usage of new equipment and programming frameworks, or the redesigning of existing ones, is a mind boggling process and getting all the more so as PC applications and systems are relied upon to interface and trade more prominent measures of data. Less programming applications are relied upon to "remain solitary" and not be associated with different applications or databases. And a conventional implementation of project in python language is quite a challenge , in this model we have used four main modules to reach the goal they are as follows,

- Data Exploration
- Pre-Processing
- Feature Engineering
- Model Selection

These are the fundamental four modules which is utilized by convolutional neural systems, YOLO-based Convolutional Neural Network group of models for object discovery and the latest variety called YOLOv3

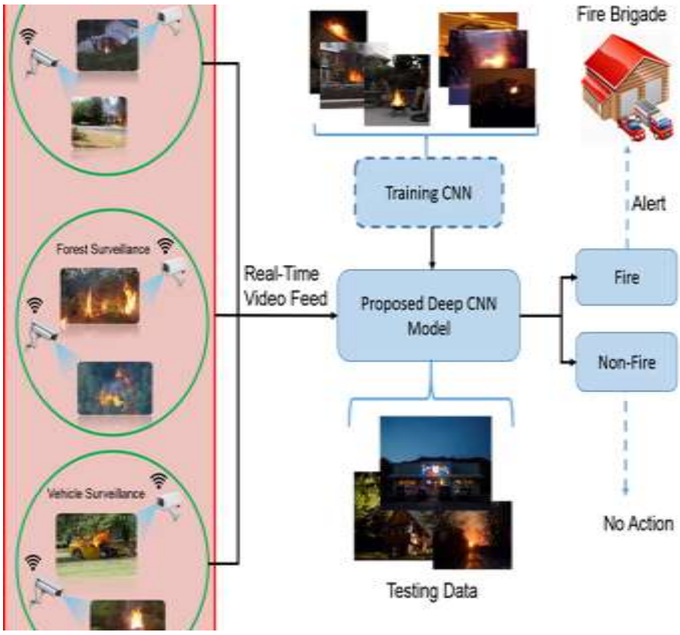

Figure 1: System Design Architecture

Implementation is that the stage of the project when the theoretical design is became a working system. Thus it are often considered to be the foremost critical stage in achieving a successful new system and in giving the user confidence that the new system will work and be effective. There are different Features and Benefits in actualizing the python language in the system.

4. MODULES

Detailed description about modules for understanding the system design and working

Module 1 : Data exploration

Data exploration, otherwise called exploratory information examination, gives a lot of straightforward devices to accomplish an essential comprehension of a dataset. The consequences of information investigation can be very helpful in getting a handle on the structure of the information, the circulation of the qualities, nearness of extraordinary qualities, and interrelationships inside the dataset. The needed data for the machine to learn is explored completely using the data exploration method in CNN. A bit of the customary estimations used are mean, standard deviation, and association.

Module 2 : Pre-processing

Pre-processing is a common name for process with images. The aim of pre-processing is an improvement of the image data that remove unwanted distortions or noise in the image some image features important for further processing. Four categories of image pre-processing methods according to the size of the pixel neighbourhood that is used for the calculation of a new pixel brightness:

1. pixel brightness transformations,
2. geometric transformations,
3.pre-processing methods that use a local neighbourhood of the processed pixel, and
4. image restoration that requires knowledge about the entire image. Other classifications of image pre-processing methods exist. Image pre-processing methods use the considerable processing in images. Neighbouring pixels indicate to one object in real images have essentially the same or similar.

Module 3 : Feature Engineering

Feature engineering is the process of transforming raw data into features that better represent the underlying problem to the predictive models, resulting in improved model accuracy on unseen data. Feature engineering turn your inputs into things the algorithm can understand. As a subfield of advanced sign handling, computerized picture preparing has numerous focal points over simple picture preparing. It permits a lot more extensive scope of calculations to be applied to the information — the point of advanced picture preparing is to improve the picture information (highlights) by stifling undesirable bends as well as upgrade of some significant picture includes so our AI-Computer Vision models can profit by this improved information to chip away it.

Module 4 : Model Selection

Model Selection is the way toward choosing one last AI model from among a group of available models in CNN of an applicant AI models for the preparation of dataset. Model determination is a procedure that can be applied both across various kinds of models (for example calculated relapse, SVM, KNN, and so on.) and across models of a similar kind designed with various model hyperparameters (for example various portions in a SVM). All models have some prescient mistake, given the factual commotion in the information, the deficiency of the information test, and the confinements of each unique model sort. Along these lines, the idea of an ideal or best model isn't helpful. Rather, we should look for a model that is "sufficient."

5. WORKFLOW

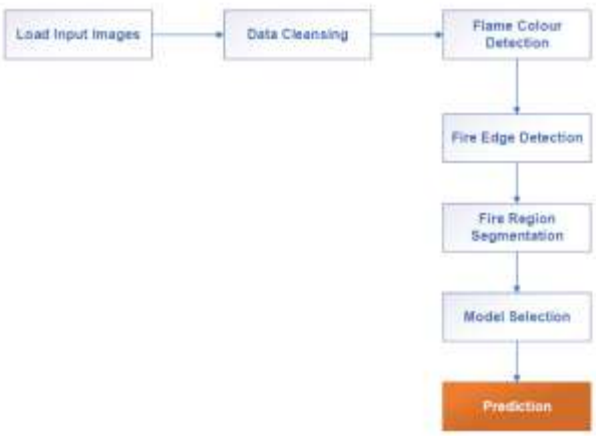

Figure 2: Workflow Model

6. CNN based Fire Detection

CNN utilizes a few highlights of the visual cortex. One of the most mainstream employments of this design is picture order. The primary errand of picture grouping is acknowledgment of the info picture and the accompanying meaning of its group. Rather than the picture, the PC sees a variety of pixels. For instance, if picture size is 300 x 300. Right now, size of the cluster will be 300x300x3. Where 300 is width, next 300 is stature and 3 is RGB channel esteems. The PC is appointed an incentive from 0 to 255 to every one of these numbers. This esteem portrays the power of the pixel at each point.The nonlinear layer, is added after each convolution operation. It has an activation function, which brings nonlinear property. Without this property a network would not be sufficiently intense and will not be able to model the response variable .

- The pooling layer, follows the nonlinear layer. It works with width and tallness of the picture and plays out a downsampling procedure on them. Accordingly the picture volume is diminished. This implies if a few highlights have just been distinguished in the past convolution activity, than a definite picture is never again required for additional handling, and it is compacted to less point by point pictures.
- A Fully connected layer, This layer takes the yield data from convolutional systems. Appending a completely associated layer as far as possible of the system brings about a N dimensional vector, where N is the measure of classes from which the model chooses the ideal class.

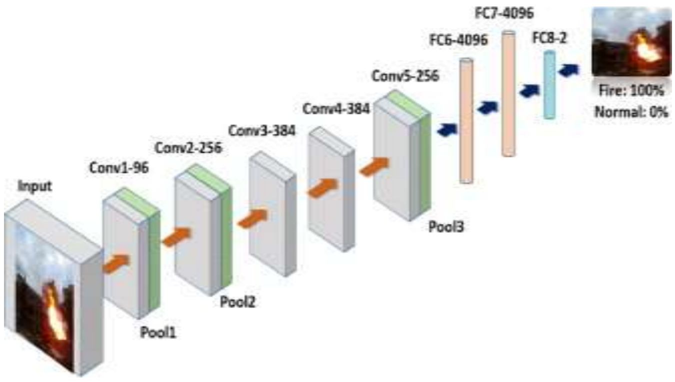

Figure 3: Shows the Layers in CNN

6.1 TRAINING AND TESTING OF DATASETS

Training plays a crucial role in CNN method, the datasets are trained in this phase in CNN , in this model the datasets are images and videos, an huge amount of images related to fire and non-fire are collected and trained , this phase requires certain amount of time to train the data with respect to the amount of available datasets, Testing proves that our trained model works right or not. Training a neural network typically consists of two phases:

1. A **forward** phase, where the input is passed completely through the network.
2. A **backward** phase, where gradients are backpropagated (backprop) and weights are updated.

While Training we consider a main parameter called **epochs,** One Epoch is when an entire dataset is passed forward and backward through the neural network only once. An epoch is comprised of one or more batches. For example, as above, an epoch that has one batch is called the batch gradient descent learning algorithm. The following figure shows the prediction accuracy for 1 epoch.

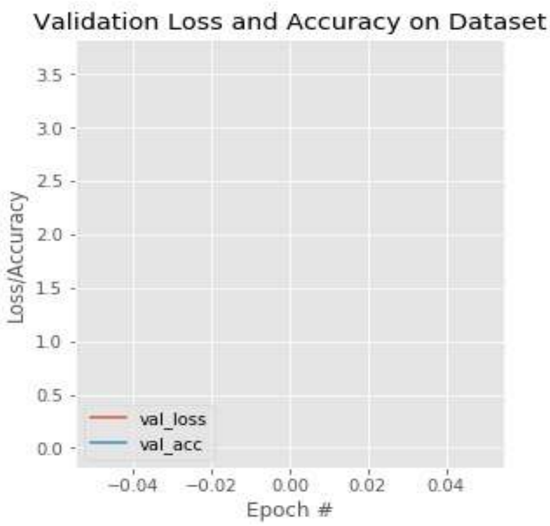

Figure 4: Accuracy for one epoch

Thus , going for single epoch value does not lead to any good prediction and hence we increase the epoch value for better prediction thus reduces the false alarm rate and improves system efficiency.

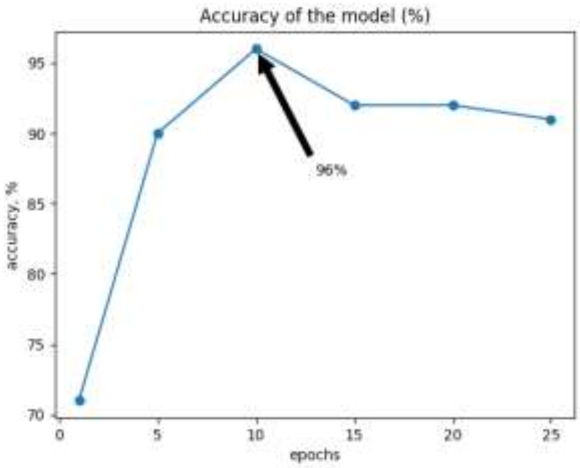

Figure 5: Accuracy for 10 epochs

It can be seen that the high accuracy (96%) is achieved after 10 epochs. That increases the prediction accuracy. But it takes considerable amount of time , we practiced it with Nvidia GPU 1050ti ddr5 4gb and AMD fx-6300 six-core processor took more than an hour to train.

6.2.2 DATASET SETUP

Dataset is an important part of machine learning technique Since it cannot able to perform any work without dataset, CNN needs more information or data in order to yield best prediction, So we use VGG16 and ResNet50.

- **VGG16** is a convolution neural net (CNN) engineering It is viewed as one of the fantastic vision model design till date. Most one of a kind thing about VGG16 is that as opposed to having an enormous number of hyper-parameter they concentrated on having convolution layers of 3x3 channel with a walk 1 and constantly utilized same cushioning and maxpool layer of 2x2 channel of walk 2. It follows this course of action of convolution and max pool layers reliably all through the entire engineering. At last it has 2 FC(fully associated layers) trailed by a softmax for yield. The 16 in VGG16 alludes to it has 16 layers that have loads. This system is a really enormous system and it has around 138 million (approx) parameters.
- **ResNet-50** is a profound remaining system. The "50" indicate to the quantity of layers it has. It's a subclass of convolutional neural systems, with ResNet most prevalently utilized for picture arrangement. ResNet-50 is a convolutional neural system that is prepared on in excess of a million pictures from the ImageNet database. The system is 50 layers profound and can order pictures into 1000 item classifications.

Thus, these pre-build datasets helps the CNN to acquire more knowledge about the system and acquires more training, We used about 864 images since the hardware requirements met only that much amount of data to yield good results. It is important to note that the image_size should be the same for all images (244*244). Different size of images result in error.

7. PROPOSED FRAME WORK

7.1 Existing System

CNN has shown great performance over the image processing technique and for prediction over the image related dataset, but the result produced by the existing system is based on the image classification and image segmentation, still there is an issue of false alarm however training more datasets may increase the accuracy of the system but still some situation exists like unknown environment and lack of data on that area in order to overcome this issue lots of advancement have made and cost of initial setup increased but only a slight decrease of false alarm. But there is also an well-known method exist sensor based fire detection system, but it takes a huge resource to setup the sensors all over the area for an instance 1000 sq.km forest region requires lots of sensors and a node center required at each and every junction in order to maintain the information and there are lots of circumstances where fire may or may not happen and hence both the method have their own issue and maintenance cost is high in order to overcome all this issue we propose a new system which will compromise the existing system.

7.2 Detailed view of Proposed framework

As earlier we prosed a new system at the beginning of the paper, You Only Look Once (YOLOv3), This is a new concept that we integrated along with CNN architecture using Tensorflow and keras , Object detection is a task in computer vision that involves identifying the presence, location, and type of one or more objects in a given image. In order to perform YOLO in the system basic setup or model for YOLO needed to be build , the following points explains the model.

1. YOLO for Object Detection
2. Experiencor YOLO3 Project
3. Object Detection With YOLOv3

7.2.1 YOLO for Object Detection

Object recognition is a systems vision task that includes both restricting at least one articles inside a picture and ordering each article in the picture. The methodology includes a solitary profound convolutional neural system (initially a rendition of GoogLeNet, later refreshed and called DarkNet dependent on VGG) that parts the contribution to a matrix of cells and every phone legitimately predicts a bouncing box and item arrangement. The outcome is an enormous number of up-and-comer jumping confines that are combined to a last expectation by a post-handling step.

7.2.2 Experiencor YOLO3 for Keras Project

Source code for every adaptation of YOLO is accessible, just as pre-prepared models. The authority DarkNet GitHub store contains the source code for the YOLO forms referenced in the papers. That gives a bit by bit instructional exercise on the best way to utilize the code for object location. Rather than building up this code without any preparation, we can utilize an outsider execution. There are some externnal executions intended for utilizing YOLO with Keras, and none seem, by all accounts, to be institutionalized and intended to be utilized as a library.

7.2.3 Object Detection With YOLOv3

The keras-yolo3 algorithm gives a great deal of capacity to utilizing YOLOv3 models, including object identification, move learning, and processing new models without any preparation. Right now, we will utilize a pre-processed model to perform object discovery. This accessible in a solitary Python document in the database called "yolo3_one_file_to_detect_them_all.py" that has around 435 lines. This content is, indeed, a program that will utilize pre-processed loads to set up a model and utilize that model to perform object identification and yield a model. It likewise relies on OpenCV. Rather than utilizing this program straightforwardly, we will reuse components from this program and build up our own contents to initially get ready and spare a Keras YOLOv3 model, and afterward load the model to make a forecast for a newimage.

8. EXPERIMENTAL RESULTS AND DISCUSSION

In this section, experimental result that means outcome of the project or system is explained, in order to perform machine learning technique an efficient system or desktop is required, we practiced the experiment with the system requirement of AMD fx6300 six core processor and a GPU of 4GB ddr5 NVIDIA 1050 ti, still our system is not efficient for this experiments , best performing GPU is needed to training the model much more faster.

Figure 6: Sample Input Images for fire, smoke and default

The above figure is the sample dataset which is not processed and it is extracted from the video file, these data are processed through CNN and YOLOv3 model that we created along with the CNN architecture.

Figure 7: Result of the System using YOLOv3

The above figure shows the experimental output, Since we trained the model about how actually a fire looks like with lots of datasets, YOLOv3 understand the object and predicted the fire, or smoke or default in the given input. Thus, reduces the false alarm. After the detection of the fire or smoke we designed our system to send a mail to nearby police station , fire department and hospital to avoid more casualty and sustain the incident.

9. CONCLUSION AND FUTURE ENHANCEMENT

In our project, Still picture based fire discovery profound learning model depending on YOLOv3 has been introduced. In view of the lightweight structure, great location capacity, and information enlargement usefulness of YOLOv3, the proposed strategy accomplishes great fire recognition execution with lower preparing intricacy and lower run time. Right now, fire and smoke video discovery strategy utilizing YOLOv3 is proposed. The pictures are partitioned into squares. The surface highlights of each picture square are processed with dim level co-event lattices. From the examination of figuring results, we see that the three surface highlights smoke do have evident contrasts with none smoke pictures. Backpropagation neural system is utilized for the separating model. Trials show that the proposed strategy can recognize Fire and smoke recordings from none fire and smoke recordings with brisk alarm and low bogus caution rate.

The calculation can give right alert the majority of time, yet there exists some bogus cautions. The fire checking scenes have confound foundation, and may have different surface component esteems. So Neural Network ought to prepare first as per the foundation and aggravations in the scenes.

Also, it is noticed that a decent video quality is required and dataset to be of same sizes.

10. REFERENCES

[1] Khan Muhammad ; Jamil Ahmad ; Zhihan Lv ; Paolo Bellavista ; Po Yang, "Efficient Deep CNN-Based Fire Detection and Localization in Video Surveillance Applications", (Volume: 49, July 2019)

[2] Tian Qiu ; Yong Yan ; Gang Lu, "An Autoadaptive Edge-Detection Algorithm for Flame and Fire Image Processing", (Volume: 61, May 2012)

[3] Ha Dai Duong ; Dao Thanh Tinh, "An efficient method for vision-based fire detection using SVM classification", on March 05,2015

[4] Jing Yang ; Feng Chen ; Weidong Zhang, "Visual-Based Smoke Detection Using Support Vector Machine", on November 07,2008

[5] Khan Muhammad ; Salman Khan ; Mohamed Elhoseny ; Syed Hassan Ahmed, "Efficient Fire Detection for Uncertain Surveillance Environment", (Volume: 15, May 2019)

[6] B. C. Ko, K.-H. Cheong, and J.-Y. Nam, "Fire detection based on visionsensor and support vector machines," Fire Safety J., vol. 44, no. 3

[7] K. Muhammad, J. Ahmad, and S. W. Baik, "Early fire detection using convolutional neural networks during surveillance for effective disaster management," Neurocomputing, vol. 288, May 2018.

[8] T. Çelik and H. Demirel, "Fire detection in video sequences using ageneric color model," Fire Safety J., vol. 44, no. 2, pp. 147–158, 2009.

[9] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in Proc. Adv. Neural Inf Process. Syst., 2012.

[10] D. Han and B. Lee, "Development of early tunnel fire detection algorithmusing the image processing," in Proc. Int. Symp. Vis. Comput., 2006.