Forest From Fire Detection (Fire Detection Model using Deep Learning)

Saumojit Nandy (saumojitnandy95@gmail.com)

Abstract:- **Every year, thousands of forest fire across the globe cause disasters beyond measure and description. There are a huge amount of very well studied solutions available for testing or even ready for use to resolve this problem. People are using sensors to detect the fire. But this case is not possible for large acres of forest. In this paper, we proposed a new approach for fire detection, in which modern technologies are used. In particular, we proposed a platform of Artificial Intelligence. The computer vision methods for recognition and detection of smoke and fire, based on the still images or the video input from the cameras. Deep learning method "convolution neural network" can be used for finding the amount of fire. This will enable the video surveillance systems on forest to handle more complex situations in real world. The accuracy is based on the algorithm which we are going to use and the datasets and splitting them into train set and test set.**

Index Terms:- Fire detection, image classification, OpenCV, deep learning, and Convolutional Neural Networks

1. INTRODUCTION

Forests are the protectors of earth's ecological balance. Unfortunately, the forest fire is usually observed when it has already spread over a large area, making its control and stoppage arduous and is impossible at times. The result is devastating loss and irreparable damage to the environment and atmosphere (30% of carbon dioxide (CO_2) in the atmosphere comes from forest fires), in addition to irreparable damage to the ecology (huge amounts of smoke and carbon dioxide (CO_2) in the atmosphere). The conventional method is to prevent illegal logging. The goal of the system is to identify the possible dangers by continuously recording the noise in the forest, by processing segments of the recorded signals and decide upon the nature of each of these segments.

It is important to move adequate fire equipment and qualified operational manpower as fast as possible to the source of the fire. Furthermore an adequate logistical infrastructure for sufficient supply with extinguishing devices and maintenance is necessary as well as continuous monitoring of fire spread. An integrated approach for forest fire detection and suppression is based on a combination of different detection systems depending on wildfire risks, the size of the area and human presence, consisting of all necessary parts such as early detection, remote sensing techniques, logistics, and training by simulation, and firefighting vehicles

Nowadays, Wireless Sensor Networks (WSNs) are critical components of the increasingly common IoT (Internet of Things) systems. Such systems have a large applicability, and the environmental monitoring field can also benefit from their innovation. An intelligent forest environment monitoring solution is based on the Raspberry Pi Model 3, analogical and digital sensors and signals analysis algorithms. Parameters such as temperature, gas concentrations, soil humidity etc. are monitored with sensors while background sounds are analyzed. Forest fire automation incorporates blocks of the functional circuit and sensor modules that are assembled as a unit thus, functioning to monitor the circumstances of a particular forest. WSN has biggest contributions since 33% researcher using WSN to tracking application, 41% use the WSN as a data exchange in their system, and 48% used WSN as data transmission between sensor nodes. A robust AdaBoost (RAB) classifier is proposed to improve training and classification accuracy.The neural network, which has 60 million parameters and 650,000 neurons, consists of five convolutional layers, some of which are followed by max-pooling layers, and three fully-connected layers with a final 1000-way softmaxn.

threefully-connected layers with a final 1000-way softmaxn. To reduce overfitting in the fully-connected layers we employed a recently-developed regularization method called "dropout" that proved to be very effective. comparison of various machine learning techniques such as regression, decision trees, neural networks etc. has been done for prediction of forest fires A semi supervised rule based classification model is rarely used to detect whether its zone is high active, medium active (MA) or low active (LA) cluster in the forest.To avoid uncontrollable wide spreading of forest fires it is necessary to detect fires in an early state and to prevent the propagation.

In order to fight against these disasters, it is necessary to adopt a comprehensive, multifaceted approach that enables a continuous situational awareness and instant responsiveness. In this paper, we proposed a new approach for fire detection, in which modern technologies are used. In particular, we proposed a platform of Artificial Intelligence. The computer vision methods for recognition and detection of smoke and fire, based on the still images or the video input from the cameras. Deep learning method "convolution neural network" can be used for finding the amount of fire.

2. RELATED WORK

2.1 In conventional fire detection, much research has continuously focused on finding out the salient features of fire images. Chen analyzed the changes of fire using an RGB and HSI color model based on the difference between consecutive frames and proposed a rule-based approach for fire decision. Celik and Demirel proposed a generic rule-based flame pixel classification using the YCbCr color model to separate chrominance components from luminance ones. In addition, Wang extracted the candidate fire area in an image using an HSI color model and calculated the dispersion of the flame color to determine the fire area. However, color-based fire detection methods are generally vulnerable to a variety of environmental factors such as lighting and shadow. Borges and Izquierdo adopted the Bayes classifier to detect fires based on additional features such as the area, surface, and boundary of the fire area to color. Mueller proposed the neural network- based fire detection method using optical flow for the fire area. In the method, two optical flow models are combined to distinguish between fire and dynamically moving objects. In addition, Foggia proposed a multi-expert system which combines the analysis results of a fire's color, shape, and motion characteristics. Although insufficient, the supplementary features to color, including texture, shape, and optical flow, can reduce the false detections. Nevertheless, these approaches require domain knowledge of fires in captured images essential to explore hand-crafted features and cannot reflect the information spatially and temporally involved in fire environments well. In addition, almost all methods using the conventional approach only use a still image or consecutive pairs of frames to detect fire. Therefore, they only consider the short-term dynamic behavior of fire, whereas a fire has a longer-term dynamic behavior.

2.2 DEEP LEARNING-BASED APPROACH

Recently, deep learning has been successfully applied to diverse areas such as object detection/classification in images, speech recognition, and natural language processing. Researchers have conducted various studies on fire detection based on deep learning to improve performance. The deep learning approach has several differences from the conventional computer vision-based fire detection. The first is that the features are not explored by an expert, but rather are automatically captured in the network after training with a large amount of diverse training data. Therefore, the effort to find the proper handcrafted features is shifted to designing a proper network and preparing the training data. Another difference is that the detector/classifier can be obtained by training simultaneously with the features in the same neural network. Therefore, the appropriate network structure becomes more important with an efficient training algorithm. Sebastien [12] proposed a fire detection network based on CNN where the features are simultaneously learned with a Multilayer Perceptron (MLP)-type neural net classifier by training. Zhang et al. [13] also proposed a CNN-based fire detection method which is operated in a cascaded fashion. In their method, the full image is first tested by the global image-level classifier, and if a fire is detected, then a fine-grained patch classifier is used for precisely localizing the fire patches. Muhammad et al. [14] proposed a fire surveillance system based on a fine-tuned CNN fire detector. This architecture is an efficient CNN architecture for fire detection, localization, and semantic understanding of the scene of the fire inspired by the Squeeze Net [15] architecture.

In the deep layer of CNN, a unit has a wide receptive field so that its activation can be treated as a feature that contains a large area of context information. This is another advantage of the learned features with CNN for fire detection. Even though CNN showed overwhelmingly superior classification performance against traditional computer vision methods, locating objects has been another problem. In the proposed method, we adopt the object detection model to localize the SRoFs and non-fire objects, which includes the flame, smoke for the SRoFs, and other objects irrelevant to the fire for the non-fire objects. The objects irrelevant to the fire increase false alarms due to variations in shadows and brightness, and will often detect objects such as red clothes, red vehicles, or sunset. We detect the fire objects by using the Faster R-CNN model, even though it does not have to be confined to the object detection model. The deep object detector, either single- or multi-stage, is usually composed of CNN-type feature extractors, followed by a localizer with a classifier. Therefore, our object detection model includes the feature extractor with a relatively wider area of receptive field than the detected SRoF area and can gather more context information. Although the CNN-based approaches provide excellent performance, it is hard to capture the dynamic behavior of fire, which can be obtained by recursive-type neural networks (RNN). LSTM proposed by Hochreiter and Schmidhuber [16] is an RNN model that solves the vanishing gradient problem of RNN. LSTM can accumulate the temporal features for decision making through the memory cells which preserve the internal states and the recurrent behavior. However, the number of recursions is usually limited, which makes it difficult to capture the long-term dynamic behavior necessary to make a decision. Therefore, special care must be taken to consider the decision based on long-term behavior with LSTM. Recently, Hu et al. [17] used LSTM for fire detection, where the CNN features are extracted from optical flows of consecutive frames, and temporally accumulated in an LSTM network. The final decision is made based on the fusion of successive temporal features. Their approach, however, computes the optical flow to prepare the input of CNN rather than directly using RGB frames.

This research paper adopts the strengths of the real-time fire detection methods reviewed using convolutional neural networks on video sequence frames and integrates that with proximity detection of fires using USB camera that fuse all the training and test data offire signatures for early detection and alerts notification while providing navigational aid to the scene of fire for the fire rescue team to respond appropriately. This novelty in the choice of using camera and video analysis overcomes the inherent problems highlighted in the review of fire detection systems.

3. SYSTEM DESIGN AND DEVELOPMENT

3.1. SYSTEM OVERVIEW

The general overview of the hardware module design and software implementation of fire detection system is shown in figure1. The hardware components of the fire detection unit, as shown in figure 2. A webcam is a video capture device that is connected to a computer or computer network, often using a USB port for video links, permitting computers to act as videophones or videoconferencing stations. Webcams can also be used with various computer video telecommunication programs which include the security surveillance and the recording of video files.

At a high level, it comprises USB camera and communication with open CV module connected to a Arduino Uno that runs the Convolutional Neural Network (ConvNet/CNN), a Deep Learning algorithm for fire detection.

The microcontroller polls the sensors at a regular interval and runs the inputs through the CNN application. If it concludes that a fire has been detected, a fire alert message is sent out through th the management information systems (MISs) to the occupants of the premises and the nearest fire station. If sending the message over the data connection is unsuccessful, then it sends the message out via short message service (SMS). The fire detection unit comprises the physical components, including the USb camera, the arduino microcontroller board, and the software that embodies the CNN fire detection algorithm and essentially drives the system.

The software subsystem is that nonphysical part of the fire detection unit, which is concerned with reading inputs from the web camera, determining whether the readings are indicative of a fire or not, using Image processing with open CV and raising alerts in cases of fires. OpenCV (Open Source Computer Vision) is a library of programming functions mainly aimed at real-time computer vision and it is library used for Image Processing. It is mainly used to do all the operation related to Images. Machines are facilitated with seeing everything, convert the vision into numbers using Pixels.

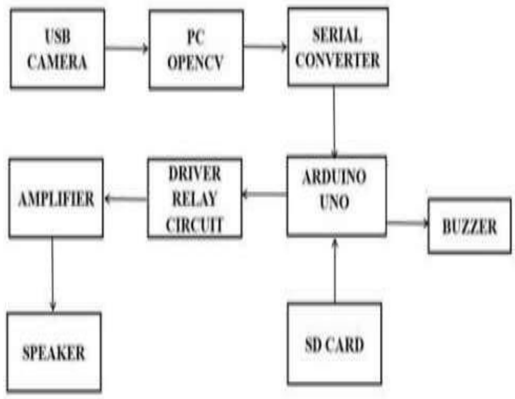

Figure1. Block Diagram of Fire detection system

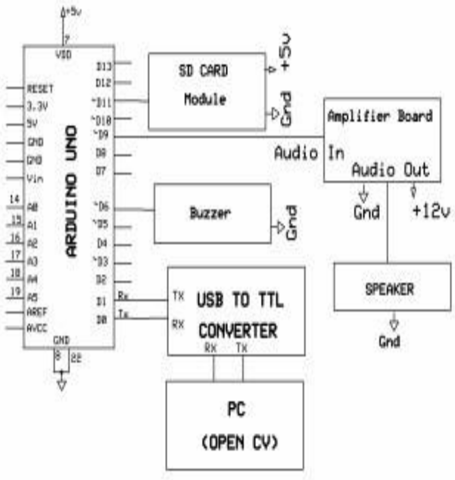

Figure 2. Circuit diagram of Hardware circuit

3.1.2.CONVOLUTIONAL NEURAL NETWORKS:

Convolutional Neural Network (ConvNet/CNN) is a Deep Learning algorithm which can take in an input image, assign importance (learnable weights and biases) to various aspects/objects in the image and be able to differentiate one from the other. The pre-processing required in a ConvNet is much lower as compared to other classification algorithms. While in primitive methods filters are hand-engineered, with enough training, CNN have the ability to learn these filters/characteristics. Figure 3. shows the structure of CNN. The layers of the network are made up of multiple three-dimensional planes. Each3-D planes consists of several neurons that make CNNs suitable for handling image data. Input layer in CNN should contain image data and it is represented by three dimensional matrix. A part of image is connected to Convo layer called feature extractor layer to perform convolution operation and calculating the dot product between receptive field and the filter. Pooling layer is used to reduce the spatial volume of input image after convolution and it is used between two convolution layers. it has two hyper parameters — Filter(F) and Stride(S). Fully connected layer involves weights, biases, and neurons. It connects neurons in one layer to neurons in another layer. It is used to classify images between different categories by training. Softmax or Logistic layer is the last layer of CNN. It resides at the end of FC layer. Logistic is used for binary classification and softmax is for multiclassification.

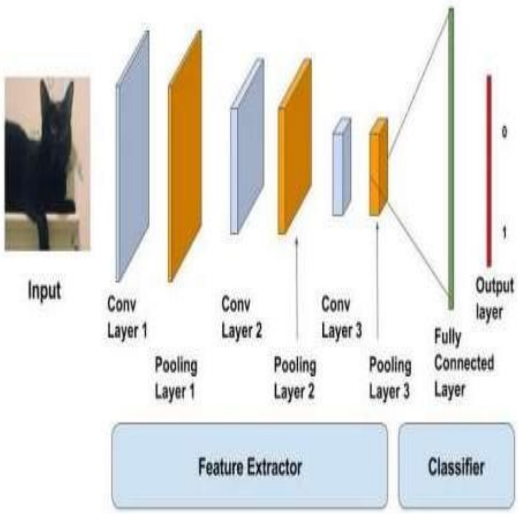

Fig3. Structure of CNN

4. SYSTEM ARCHITECTURE SYSTEM IMPLEMENTATION AND TESTING

In convolution operation, several kernels of different sizes are applied on the input data to generate feature maps. These features maps are input to the next operation known as subsampling or pooling where maximum activations are selected from them within small neighborhood. These operations are important for reducing the dimension of feature vectors and achieving translation invariance up to certain degree. Another important layer of the CNN pipeline is fully connected layer, where high-level abstractions are modeled from the input data. Among these three main operations, the convolution and fully connected layers contain neurons whose weights are learnt and adjusted for better representation of the input data during training process

The system architecture consists of both hardware and software components. hardware components comprise of the fire detection system, which will have the installation of the software components. For enhanced fire detection, a surveillance camera unit is incorporated as part of the hardware system implementation, which will continuously monitor the premises and send the video feed to a centralized server for fire incident detection and alert notification

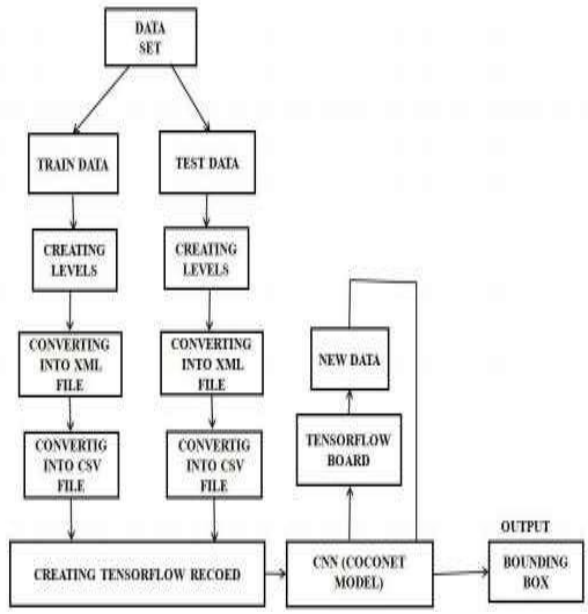

Figure 4: Flow diagram for fire detection software

The object detector is created to train a robust classifier. we need a lot of pictures which should differ a lot from each other. So they should have different backgrounds, random object, and varying lighting conditions. The different samples fire classes are as shown in figure 1.

Figure 5: Sample images for classes.

we need to move about 80 percent of the images into the object_detection/images/train directory and the other 20 percent in the object_detection/images/test directory. In order to label our data, we need some kind of image labeling software. Labeling is a great tool for labeling image. To create the bounding box the "Create RectBox" button can be used. After creating the bounding box and annotating the image you need to click save. This process needs to be repeated for all images in the training and testing directory. With the images labeled, we need to create TFRecords that can be served as input data for training of the object detector. In order to create the TFRecords we will use two scripts from Data Tran's raccoon detector. In order to test our newly created object detector, we can use the code which we already created. the video stream from the surveillance camera is sent to the server where the CNN model is run on the data, and if a fire is detected, the alerting system is thereby triggered.

5. SYSTEM IMPLEMENTATION AND TESTING

The convolutional neural network was trained on over 1000 images of instances of fire and non-fire situations. The data were collected from online image data sources. Sample images used for the convolutional neural networks are shown in Figure 5. As a principle for the training of neural networks, the dataset was split into training and testing sets. To get an inference from the model, test video streams were used as input and passed through to the system. classifier outputs probabilities for the two classes: "fire" and "no fire." The class with the maximum probability score is considered as the result of the classifier. classifier module was implemented with Google's Tensor Flow. Tensor Flow is an open-source software library provided by Google for numerical computation using data flow graphs After the training the network on the data, a classifier with an accuracy of 94% was achieved. The classification process is as follows; the video feed is preprocessed, and frames are extracted from the video. The extracted frames/images are classified to determine the condition of the area

6. RESULTS AND DISCUSSIONS

The aim of this work is to present a method that can be smoothly deployed to an embedded device in order to finally build a complete fire detection unit. Therefore, it becomes inevitable to use a test dataset that includes images that are often encountered in real-world fire emergencies with an image quality that is commonly obtained with a camera attached to low- cost hardware like Arduino Uno, a microcontroller board based on the ATmega328P. The video classifier performed very well on the tests run on the classifier module. To avoid the instances of false alarms being triggered, a threshold for the classifier confidence was set. Hence, the alarm is only triggered when the confidence is greater or equal to the threshold. The aim is to detect a fire from the video stream with very high accuracy and trigger an alert as quickly as possible. To boost the speed of the classifier, TensorFlow's "optimize_for_inference" script was used to remove all unnecessary nodes in the module. The script also does a few other optimization processes like normalizing operations into the convolutional weights that help speed up the model.

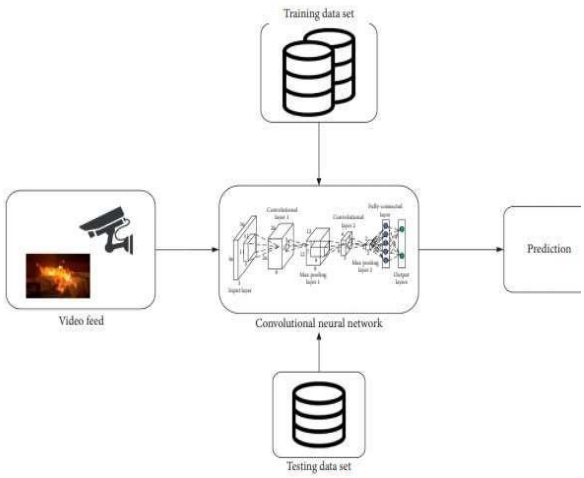

Figure 6: Video processing unit on a dedicated server.

7. CONCLUSION AND RECOMMENDATION

In this work, we present a Convolutional neural network build from scratch and trained on a very diverse dataset. The ultimate aim of the complete work is to develop an internet of things (IoT) capable fire detection unit that can effectively replace the current physical sensor based fire detectors and also can reduce the associated problems of false and delayed triggering with such fire detectors. The introduced neural network can smoothly run on a low-cost embedded device like Arduino Uno, a microcontroller board based on the ATmega328P at a frame rate of 24 frames per second. The performance obtained by the model on a standard fire dataset and a

self-made test dataset (consisting challenging real- world fire and non-fire images with image quality that is similar to the images captured by the camera attached to Arduino) in terms of accuracy. Moreover, the IoT functionality allows the detection unit to provide real-time visual feedback and fire alert in case of fire emergencies to the user. Although, this work improved the flame detection accuracy, yet the number of false alarms is still high and further research is required in this direction. In addition, the current flame detection frameworks can be intelligently tuned for detection of fire. This will enable the video surveillance systems on forest to handle more complex situations in real world.