FIRE DETECTION USING OPENCV AND G-STREAMER

Mentor:-Prof. Ayan Panja[1], Subroto Rakshit[1], Pratyush Ranjan[1], Subha Chakraborty[1], Prabhat Kumar[1], Payel Chatterjee[1], and Ajay Kumar Mandal[1]

[1]Department of CSE(AIML), Institute of Engineering Management, Kolkata, India
[1](ayan.panja@iemcal.com,rakshitsubroto123@gmail.com,p4pratyushranjan@gmail .com,suboff145@gmail.com,kumarprabhat748@gmail.com,chatterjeepayel958@gmail.com, ajaycool78700@gmail.com)

Abstract

In this modern world, AI and ML play a very important role. The market of AI and ML is increasing. All the domains are now based on AI and ML or they adapting them to AI and ML. Most of the things that we utilize in this generation are based on artificial intelligence and machine learning. We are allowing computers to think and take decisions. Now computers are self-working. They have the ability to control themselves and detect anything. So our project is also based on AI and ML. We are going the build a model which will detect smoke and fire at any distinct place and it will automatically generate an alarm for it. All the work will be computer-based. It can be utilized by any county for the fire and smoke detection system. The modules that will be used are G-streamer for the port transfer and openCV for detecting purposes. There will be cameras for detecting the fire and smoke. Signals will be shared via the internet and if the machine algorithm identifies any smoke and fire then it will send a report to the server, and it will also generate an alarm in such situations. The purpose of the G-streamer will be to break the packets and generate the report and a pipeline will be created by it and accordingly the system will work. The main purpose of G- streamer is to create a pipeline for the project. The main purpose of Open-CV is to handle the media file and to generate a report in array format and which will transferred by the pipeline that will be created using the G-streamer.

Keywords
Project, Pipeline, G-streamer, Protocols, report, open-CV, packets

1 Introduction

1.1. Multimedia over the internet:

The internet was originally designed to support data communications. Most of the traffic initially includes data such as e-mails and files. Voice traffic was served exclusively by telephone networks. As the Internet grew in terms of the number of nodes, applications, and users, the need for multimedia communication over the internet emerged. Animation, voice, and video clips are now common on the internet. Multimedia networking products like internet telephone, internet T.V, and video conferencing are available in the market. Shortly, people will enjoy other multimedia products in distance learning, distribution simulation, distributed workgroups, and other areas.

1.2. Real Time Streaming Protocol:

This project is based on Real-Time Streaming, so Real-Time Streaming Protocol(RTPS) will be used in this project. RTSP is an application-level networking protocol for a communication system that transfers real-time data from multimedia to an endpoint device by communicating directly with the server streaming the data.

This protocol controls the flow of data between the media devices and also establishes and controls the media stream between client devices and serves by serving as a network remote control for continuous media, such as audio and video. It does not stream the multimedia itself but communicates with the server that streams the multimedia data.

It also provides a protocol for downloading the media file in the device and also gives the control to user-related playing the file, i.e., play, pause, fast, slow, etc. The quality of the file can be controlled.

RTSP is a stateful protocol. A stateless protocol does not require the server to retain session information or status about each communicating partner for the duration of multiple requests. In contrast, a protocol that requires the keeping of the internal state on the server is known as a stateful protocol. In a stateful protocol, there is a tight dependency between client and server. If the client sends a request to the server then it expects some kind of response, if it does not get any response then resend the request. A stateful server keeps the state of connections.

FIGURE NEEDED HERE

Fig 1.2.1 shows the connection setup between the server and the clients.Packets are transferred between the server and the clients and a communication channel is a setup between them. After the channel is set up, flies are shared in packets. There are different types of layers in the TCP/UDP

2 Gstreamer

GStreamer is an open-source multimedia framework that is mainly used to create media applications. Such as for streaming, media playback, non-linear editing, etc. GStreamer can be used to build a system that reads files in one format, processes them, and exports them in another. The GStreamer framework is designed to make it easy to write applications that handle audio or video or both. It uses plugins that will provide various codecs and other.

Its most common uses are as follows:
- to extract audio track from some media file container,
- to transcode video and/or audio track (change encoding format) so that I can play it in the web browser,
- to change the resolution (width/height) and FPS (frames per second) of some video files,
- to convert (transmit) one media file container to another (e.g. from MP4 to MPEG TS),
- to stream my movie to another PC in a local network.

2.1. Why to use Gstreamer:

Dealing with multimedia is not an easy task. Therefore, selecting the proper framework to process audio and video streams becomes a key point to ensure a successful project. Developers facing this challenge should consider the following factors before making their choice for the right multimedia framework:
-coding complexity (architecture)
-cross-platform support
-multimedia technologies coverage (codecs, filters, etc.)
-documentation and support are available.

GStreamer excels in the above-mentioned criteria.

its intelligence plugin architecture and a comprehensive core library make your application development easy.It provides you with well-tested elements for many of your needs. It works on all major operating systems (ie. Linux, Android, Windows, Mac OS X, iOS) and runs on all major hardware architectures including x86, ARM, MIPS, SPARC. It counts with an extensive list of multimedia plugins (encoders, decoders, content filters...). It allows easy integration of 3rd party ones.

Lastly, GStreamer counts with wide and well-structured documentation available for developers and a big community whose contributions ensure a continuous improvement of this framework.

2.2. How can we use Gstreamer in our IT Projects:

So, we can use Gstreamer to
- Extract audio from media files recorded by a Surveillance camera.
- Extract video or image from media files too.

Gstreamer can handle and process media files using a variety of components (elements) that can be found in Gstreamer plugins. Components can be interconnected forming a pipeline where data usually flows upstream (from source to sink, from "left to right"). In general, elements are connected like this:-

figure needed here

And are capable to exchange messages to inform each other about some events happening during data processing (e.g. sync messages).
For extracting the media files:-
we will need to learn more about how Gstreamer forms a pipeline for media processing. The

pipeline will be very similar to the audio one with the difference that we need to use a video decoder and video encoder.

3 Open-CV

OpenCV is the huge open-source library for computer vision, machine learning, and image processing and now it plays a major role in real-time operation which is very important in today's systems. By using it, one can process images and videos to identify objects, faces, or even the handwriting of a human. When it is integrated with various libraries, such as NumPy, python is capable of processing the OpenCV array structure for analysis. To Identify image patterns and its various features we use vector space and perform mathematical operations on these features.

3.1. Applications of OpenCV:-

There are lots of applications that are solved using OpenCV, some of them are listed below:-
- face recognition,
- Automated inspection and surveillance,
- number of people – count (foot traffic in a mall, etc),
- Vehicle counting on highways along with their speeds,
- Interactive art installations.

3.2. Image Processing:-

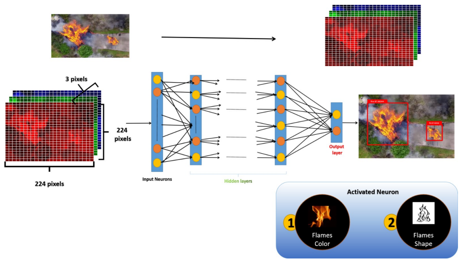
Image processing is a method to perform some operations on an image, to get an enhanced image, and or to extract some useful information from it. If we talk about the basic definition of image processing then "Image processing is the analysis and manipulation of a digitized image, especially to improve its quality.

3.3. Features of Open-CV Library:

Using OpenCV library, you can:-
– Read and write images
- Capture and save videos
- Process images (filter, transform)
- Perform feature detection

4 Visualization of Convolution Neural Network

Here is the visualization of the CNN



5 Different Neural Networks used in our project

1. ZFNet:
ZFNet, short for "Zeiler and Fergus Net," is a convolutional neural network (CNN) architecture designed for image classification tasks. Developed by Matthew D. Zeiler and Rob Fergus in 2013, ZFNet gained prominence for winning the ImageNet competition. It features a deep architecture with multiple convolutional layers and max-pooling, effectively capturing hierarchical features in images. ZFNet played a crucial role in advancing the field of deep learning for computer vision, paving the way for subsequent architectures like AlexNet and influencing the design of modern CNNs.

2. AlexNet:
AlexNet is a pioneering convolutional neural network (CNN) architecture that significantly contributed to the advancement of deep learning in computer vision. Developed by Alex Krizhevsky, Ilya Sutskever, and Geoffrey Hinton, the model won the 2012 ImageNet Large Scale Visual Recognition Challenge. Introduced in 2012, AlexNet consists of eight layers, including five convolutional layers and three fully connected layers. It employs rectified linear units (ReLU) as activation functions and utilizes techniques such as dropout for regularization. AlexNet's success marked a breakthrough in image classification tasks, leading to the widespread adoption of deep neural networks in various applications.

3. Basic Model:
In the realm of machine learning, a "basic model" typically denotes a simple algorithm or model serving as an initial framework for understanding and problem-solving. These rudimentary models, often straightforward in design, help lay the groundwork for more sophisticated approaches. Basic models can include linear regression for predicting relationships in data or elementary decision trees for classification tasks. They serve as essential starting points for learning key concepts before progressing to more complex techniques. These foundational models are crucial in building a solid understanding of ma-

chine learning principles and are often employed in educational settings and introductory studies within the field.

4. Inception V3:

Inception v3 is a convolutional neural network (CNN) architecture designed for image classification and object recognition. Developed by Google's research team, it is a successor to the original Inception model. Introduced in 2015, Inception v3 features advanced architectural improvements, including factorized convolutions and auxiliary classifiers during training. With approximately 48 layers, it efficiently captures intricate hierarchical features in images, demonstrating high accuracy on tasks like the ImageNet Large Scale Visual Recognition Challenge. Inception v3 has become a prominent model in computer vision, known for its effectiveness in balancing computational efficiency and performance in deep learning applications.

5. Efficient Net:

EfficientNet is a groundbreaking convolutional neural network (CNN) architecture designed to achieve superior performance with fewer parameters and computational resources. Introduced by Google researchers in 2019, EfficientNet leverages compound scaling, systematically balancing model depth, width, and resolution to optimize efficiency. By dynamically scaling these dimensions, EfficientNet achieves state-of-the-art results in image classification tasks while maintaining a more efficient use of resources compared to traditional architectures. Its innovative approach to scaling has made EfficientNet a widely adopted model for various computer vision applications, offering an excellent trade-off between accuracy and computational efficiency in deep learning.

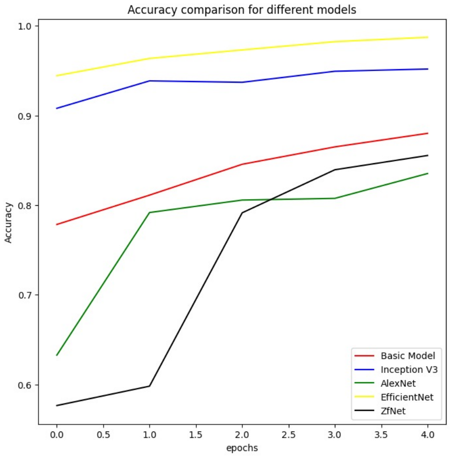
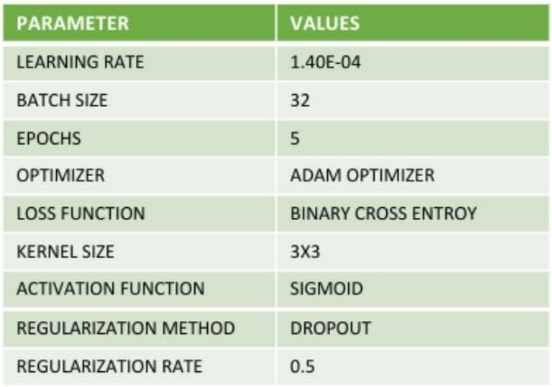


Figure 1: Accuracy plot



PARAMETER	VALUES
LEARNING RATE	1.40E-04
BATCH SIZE	32
EPOCHS	5
OPTIMIZER	ADAM OPTIMIZER
LOSS FUNCTION	BINARY CROSS ENTROY
KERNEL SIZE	3X3
ACTIVATION FUNCTION	SIGMOID
REGULARIZATION METHOD	DROPOUT
REGULARIZATION RATE	0.5

Figure 2: The Hyperparameters involved in our model is listed in the above table

6 Their Accuracy

Their ROC Curve and Their Accuracy

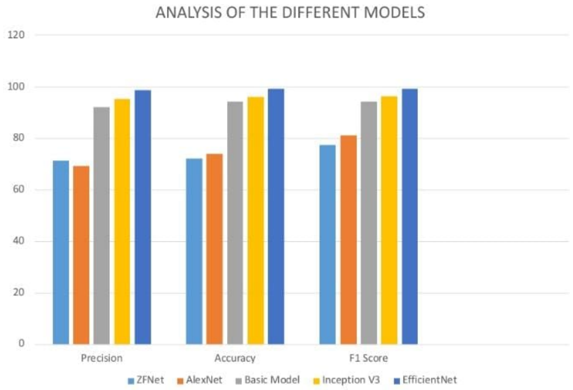


Figure 3: Bar Graph

7 Conclusion

The project deals with real time detection of forest fire and smoke.

The video information from local CCTVs and other camera sources is taken, on which predictions is made using a CNN model, to detect forest fire.

Efficient-Net is chosen as the predicting model, after trying multiple models, it has the highest **precision(98.71), accuracy(99.1) and F1 Score(99.12)** as compared to other models.

This system can be integrated with pre-existing warning systems(ex:- SMS warning, etc), and will aid authorities to know about forest fire faster, thereby ensuring the fire will be dealt with as soon as possible.

8 References:-

Authors of the said project: Robert A. Sowah, Abdul R. Ofoli, Selase N. Krakani, Seth Yayra Fiawoo
https://ieeexplore.ieee.org/abstract/document/7575685
Authors: Yi-Chun Chen, Halim Fathoni, Chao-Tung Yang
https://ieeexplore.ieee.org/abstract/document/9302759/authorsauthors
Authors: Sergio Saponara, Abdussalam Elhanashi, Alessio Gagliardi
Link:https: //link.springer.com/article/10.1007/s11554-020-01044-0
http://4youngpadawans.com/gstreamer-real-life-examples/
https://gstreamer.freedesktop.org/documentation/installing/on-windows.html?gi-language=c
https://fluendo.com/en/what-is-gstreamer/