

**EMERGING METHODS FOR EARLY DETECTION**

**OF FOREST FIRES**

## A PROJECT REPORT

### Submitted by

# GOKULRAJ R

# HARIPRIYAA VT

# HARI SHRUTHI TK

# HARIPRASATH AS

### in partial fulfillment for the award of the degree of

# BACHELOR OF ENGINEERING

**IN**

ELECTRONICS AND COMMUNICATION ENGINEERING

**SRI KRISHNA COLLEGE OF ENGINEERING AND TECHNOLOGY**

**(An Autonomous Institution, Affiliated to Anna University Chennai - 600 025)**

#### NOVEMBER 2022



# BONAFIDE CERTIFICATE

Certified that this project report **“Emerging Methods for Early Detection of Forest Fires”**

is the bonafide work of “**GOKULRAJ R(19EUEC046), HARIPRIYAA VT(19EUEC047)**

**HARI SHRUTHI TK(19EUEC048), HARIPRASATH AS(19EUEC049)”**who carried out the project work under my supervision.

<<Signature of the Head of the Department>> <<Signature of the Supervisor>>

##### SIGNATURE SIGNATURE

<<Name>> <<Name>>

##### HEAD OF THE DEPARTMENT SUPERVISOR

<<Academic Designation>>

<<Department>> <<Department>>

<<Full address of the Dept & College >> <<Full address of the Dept & College >>

Submitted for the Project viva-voce examination held on

INTERNAL EXAMINER EXTERNAL EXAMINER

**ABSTRACT:**

Forest fires are occurring throughout the year with an increasing intensity in the summer and autumn periods. These events are mainly caused by the actions of humans, but different nature and environmental phenomena, like lightning strikes or spontaneous combustion of dried leafs or sawdust, can also be credited for their occurrence. Regardless of the reasons for the ignition of the forest fires, they usually cause devastating damage to both nature and humans. Forest fires are also considered as a main contributor to the air pollution, due to the fact that during every fire huge amounts of gases and particle mater are released in the atmosphere. To fight forest fires, different solutions were employed throughout the years. They ware primary aimed at the early detection of the fires. The simplest of these solutions is the establishment of a network of observation posts - both cheap and easy to accomplish, but also time-consuming for the involved people. The constant evolution of the information and communication technologies has led to the introduction of a new generation of solutions for early detection and even prevention of forest fires. ICT-based networks of cameras and sensors and even satellite-based solutions were developed and used in the last decades. These solutions have greatly decreased the direct involvement of humans in the forest fire detection process, but have also proven to be expensive and hard to maintain. In this paper we will discuss and present two different emerging solutions for early detection of forest fires. The first of these solutions involves the use of unmanned aerial vehicles (UAVs) with specialized cameras. Several different scenarios for the possible use of the drones for forest fire detection will be presented and analysed, including a solution with the use of a combination between a fixed-wind and a rotary-wing UAVs.

(i)

**TABLE OF CONTENT**

|  |  |  |
| --- | --- | --- |
| CHAPTER NO | TITLE | PAGE NO |
|  | **ABSTRACT** | (i) |
|  | **LIST OF FIGURES** | (ii) |
|  | **LIST OF ABBREVATIONS** | (iii) |
| 1. | **INTRODUCTION** | **7** |
| 2. | **LITERATURE SURVEY** | **8** |
| 3. | **EXISTING SYSTEM** | **10** |
| 4. | **PROPOSED SYSTEM**  4.1 BLOCK DIAGRAM  4.2 SUMMARY OD DATASET ATTRIBUTE  4.3 CONVOLUTIONAL NEURAL NETWORK  4.4 MODEL BUILDING | **11**  **12**  **13**  **14** |

|  |  |  |
| --- | --- | --- |
| 5. | **VIDEO ANALYSING** | **21** |
| 6. | **CONCLUSION** | **27** |
| 7. | **REFERENCE** | **28** |

**LIST OF FIGURES**

(**ii)**

|  |  |
| --- | --- |
| **FIGURE NO** | **TITLE** |
| 1 | Dataset |
| 2 | Artificial Neural Network |
| 3 | Data acquisition and preprocessing |
| 4 | Model Building |
| 5 | Padding |
| 6 | Activation Function |
| 7 | Pooling Operation |
| 8 | Dense Layer |
| 9 | Training the model |
| 10 | Saving the model |
| 11 | Prediction of the model |
| 12 | Importing Libraries |
| 13 | Loading Model |
| 14 | TWILIO Service |
| 15 | Sending Alert Message |
| 16 | CNN Tensorflow |

**LIST OF ABBREVIATIONS**

**(iii)**

|  |  |
| --- | --- |
| **ABBREVIATIONS** | **EXPANSION** |
| MLP | Multilayer Preceptron |
| RBF | Radial Basis Function |
| AI | Artificial Intelligence |
| GIS | Geographic Information System |
| WSN | Wireless Sensor Networks |
| DNF | Disjunctive Normal Form |
| ANN | Artificial Neural Network |
| PKF | Polynomial Kernel Function |
| SVM | Support Vector Machine |
| LR | Logistic Regression |
| MLR | Multiple linear regression |
| RF | Random forest |
| HS | Historical Satellite |

CHAPTER 1

INTRODUCTION

Forest fires have been and still are serious problem for the European Union and for all other countries in Europe. In the year 2000, the EU has established the European Forest Fire Information system (EFFIS) [1], which will soon become part of the European Emergency Management Service, maintained by the Copernicus Earth Observation Programme [2]. This system provides valuable near real-time and also historical data on the forest fires in Europe, the Middle East and North Africa. Currently EFFIS is being used and supported with data by 25 EU member states and by numerous other countries. According to the annual report of EFFIS for 2016 [3], more than 54,000 forest fires have occurred all around Europe and they have led to nearly 376 thousand hectares of burnt areas. If we compare these values to the average values from the EEFIS reports for the period 2006-2015, the number of forest fires have decreased by 13327 or by the nearly 20%. This decease can be explained with the more severe actions and sanctions towards the arsonists and with the introduction of more advanced technical solutions for early detection of the fires. Even though their number is decreasing, the forest fire continue to be extremely devastating forest fire form 2018, which took place in the Attica region of Greece and led to more than 90 fatalities and to more than200 injured people, as well as to the destruction to thousands of buildings [4].

The most important factors in the fight against the forest fires include the earliest possible detection of the fire event, the proper categorization of the fire and fast response from the fire services. Several different types of forest fires are known, including ground fires, surface fires and crown/tree fires [5]. Each of these types of forest fires is specific and the proper counteractions against it must be considered and implemented to successfully fight it. Over the years the detection of forest fires has been conducted in different ways, ranging from the use of forest outposts to fully automated solutions.

CHAPTER 2

LITERATURE SURVEY

Burned areas in forest fires were predicted using estimation methods as the Multilayer Perceptron (MLP), SVM, Radial Basis Function (RBF) networks and fuzzy logic. The results indicate that MLP gives more accurate results. Available and Reliable Storage for an Incompletely Trusted Environment (FARSIGHT) simulator was used to predict forest fires spread in the Euro-Mediterranean countries. The outputs of FARSIGHT were obtained by two models, custom fuel model and standard fuel model. The experimental results showed that the accuracy of the custom fuel model was better than the standard fuel model. An intelligent system called geometric semantic genetic programming to predict burned areas. The results obtained using that intelligent systems were better than using standard genetic programming. A novel system called forecast to predict the spread of forest fires in the future. The forecast is a system that combines Artificial Intelligence (AI) and Geographic Information Systems (GIS). The forecast obtained more accurate results when compared to other random prediction models. A machine-learning algorithm based on Wireless Sensor Networks (WSN) to predict forest fires. A fire prediction tool called Disjunctive Normal Form (DNF) model to predict forest fires. The results obtained from the DNF model were compared with other machine learning models as naive Bayes, decision tree, SVM, RBF, and polynomial kernel functions. The DNF model gave the highest average accuracy with 97.8% among the other machine learning models. An algorithm that depends on SVM to predict forest fires. SVM used two class predictions of fire risk. The results demonstrated that the accuracy of SVM was approximately 96%. ANN model was used to predict the size of burned areas of forest fires in southern Spain. ANN was used in two stages: classifying forest fires size and evaluation of the burned surface areas. The results mentioned that the process of prediction was over 60%, prediction can reach more than 70% in some central areas. A probabilistic model was used to predict forest fires. There were three steps to design the probabilistic model. In step 1, the probabilistic model of forest fires was built from data of weather forecast and historical satellite. In step 2, the prediction of forest fires was produced using the data of the weather forecast as an input in the model of forest fires. In step 3, the warnings of forest fires were transported on different levels based on the need of the user. Machine learning models to predict the size of forest fires at the time of their inflammation. Decision trees, random forests, and MLP models were used in the process of prediction. The decision tree model predicted that 40% of the inflammation led to a large number of fires, and this per cent is about 75% of the total burned area. Random forests and MLP models were tested, but they did not perform the accuracy as the decision tree model. Different machine learning models to predict forest fires in Slovenia. Logistic regression, decision tree, random forests, bagging, and boosting of decision tree models were used to predict forest fires in Slovenia. These models were applied to these three data sets: Kras region, Primorska region, and continental Slovenia. From the experimental results, the bagging decision tree model obtained the best accuracy for all the data sets. Semiparametric models were used to predict forest fires.

CHAPTER 3

EXISTING SYSTEM

The existing system a novel system for detecting fire using Convolutional Neural Networks (CNN). Detection of fire can be extremely difficult using existing methods of smoke sensors installed in the buildings. They are slow and cost inefficient due to their primitive design and technology. This paper critically analyzes the scope of Artificial intelligence for detection and sending alerts with video from CCTV footages. This project uses self-built dataset containing video frames with fire. The data is then preprocessed and use the CNN to build a machine learning model. The test set of the dataset is given as input for validating the algorithm and experiments are noted. The project focus on building cost efficient and highly accurate machine that can be used in almost any use case of fire detection.

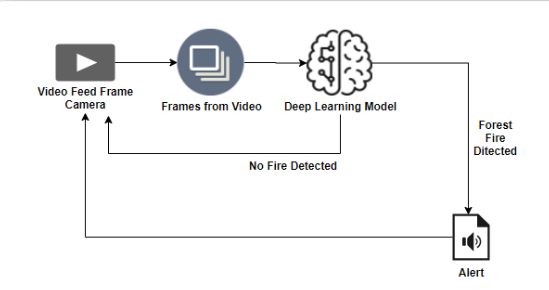
Existing system they are used Robots for Extinguish fire robot is interfaced with several sensors

* Unmanned Ariel Vehicle (UAV) – For monitoring Forest fire.
* Unmanned Ground Vehicle (UGV) – For extinguish fire.
* Both Vehicles are communicating through Radio Frequency Communication if UAV is detecting fire means it will send to UGV.
* UGV will come to that spot and extinguish fire.\

CHAPTER 4

PROPOSED SYSTEM

The aim of the project is to detect the forest fire.

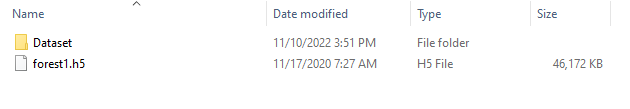
**4.1 BLOCK DIAGRAM:**

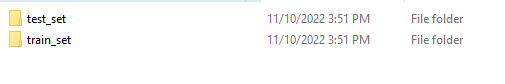
At present, most of the applications in forest fire identification are directly applied to CNN on the original image set. Due to the complex background and a number of interference in the original image, the result of the training is not so good. Therefore, in this paper, a method is proposed to segment the candidate flame region based on the color feature, and then part of the image is sent to the CNN network for training, which can extract features more specifically and improve the recognition rate of forest fire image effectively. In the training phase, firstly, the binary image of the suspected flame region is segmented, and the result obtained by performing AND operation between the binary image and the original image is used as a training set, and a label is set for each image. A network model is obtained after training the CNN according to the training set. In the testing phase, similarly, the binary image of the suspected flame region is firstly segmented, and the result obtained by performing AND operation with the original image is used as a testing set. The testing set image is sent to the trained network model to obtain the recognition result.

**4.2 SUMMARY OF DATASET ATTRIBUTE:**

The datasets are given as images and are classified as test and train datasets. Image data is most often used to represent graphic or pictorial data. The term *image*inherently reflects a graphic representation, and in the *GIS world*, differs significantly from raster data. Most often, image data is used to store remotely sensed imagery, e.g. satellite scenes or orthophotos, or ancillary graphics such as photographs, scanned plan documents, etc. Image data is typically used in GIS systems as background display data (if the image has been rectified and georeferenced); or as a graphic attribute. Remote sensing software makes use of image data for image classification and processing. Typically, this data must be converted into a raster format (and perhaps vector) to be used analytically with the GIS. Image data is typically stored in a variety of de facto industry standard proprietary formats. These often reflect the most popular image processing systems. Other graphic image formats, such as TIFF, GIF, PCX, etc., are used to store ancillary image data. Most GIS software will read such formats and allow you to display this data. Here the forest with fire and without fire are created as datasets.

FIG 1



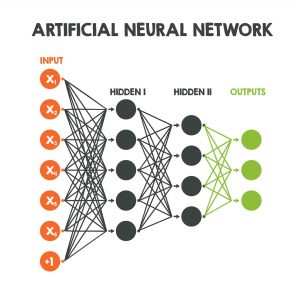




* 1. **CONVOLUTIONAL NEURAL NETWORK:**

Convolutional neural networks are distinguished from other neural networks by their superior performance with image, speech, or audio signal inputs. They have three main types of layers, which are: Convolutional layer, Pooling layer, Fully-connected (FC) layer. The convolutional layer is the first layer of a convolutional network. While convolutional layers can be followed by additional convolutional layers or pooling layers, the fully-connected layer is the final layer. With each layer, the CNN increases in its complexity, identifying greater portions of the image. Earlier layers focus on simple features, such as colors and edges. As the image data progresses through the layers of the CNN, it starts to recognize larger elements or shapes of the object until it finally identifies the intended object. The convolutional layer is the core building block of a CNN, and it is where the majority of computation occurs. It requires a few components, which are input data, a filter, and a feature map. Let’s assume that the input will be a color image, which is made up of a matrix of pixels in 3D. This means that the input will have three dimensions—a height, width, and depth—which correspond to RGB in an image. We also have a feature detector, also known as a kernel or a filter, which will move across the receptive fields of the image, checking if the feature is present. This process is known as a convolution. The feature detector is a two-dimensional (2-D) array of weights, which represents part of the image. While they can vary in size, the filter size is typically a 3x3 matrix; this also determines the size of the receptive field. The filter is then applied to an area of the image, and a dot product is calculated between the input pixels and the filter. This dot product is then fed into an output array. Afterwards, the filter shifts by a stride, repeating the process until the kernel has swept across the entire image. The final output from the series of dot products from the input and the filter is known as a feature map, activation map, or a convolved feature.

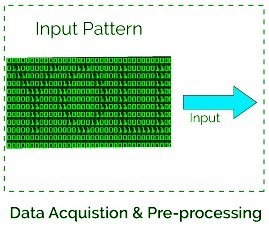
FIG 2



**IMAGE PREPROCESSING:**

Read the picture files (stored in data folder).Decode the JPEG content to RGB grids of pixels with channels. Convert these into floating-point tensors for input to neural nets.Rescale the pixel values (between 0 and 255) to the [0, 1] interval (as training neural networks with this range gets efficient).

FIG 3



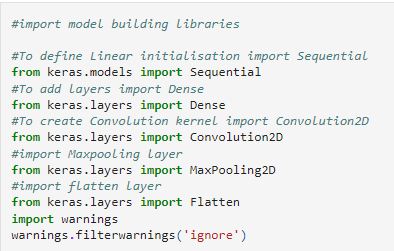
It may seem a bit fussy, but Keras has utilities to take over this whole algorithm and do the heavy lifting for you. Keras has a module with image-processing helping tools, located at keras.preprocessing.image. It contains the class *ImageDataGenerator*, which lets you quickly set up Python generators that can automatically turn image files on disk into batches of preprocessed tensors.

**4.4 MODEL BUILDING:**

**MODEL BUILDING LIBRARIES:**

These are  importing some important libraries and the modules that are required for building the convolutional model. The Conv2D layer is the convolutional layer required to creating a convolution kernel that is convolved with the layer input to produce a tensor of outputs.

FIG 4

****

**ADD CNN LAYERS**

### Filters

The primary purpose of convolution is to find features in the image using a feature detector. Then put them into a feature map, which preserves distinct features of images.

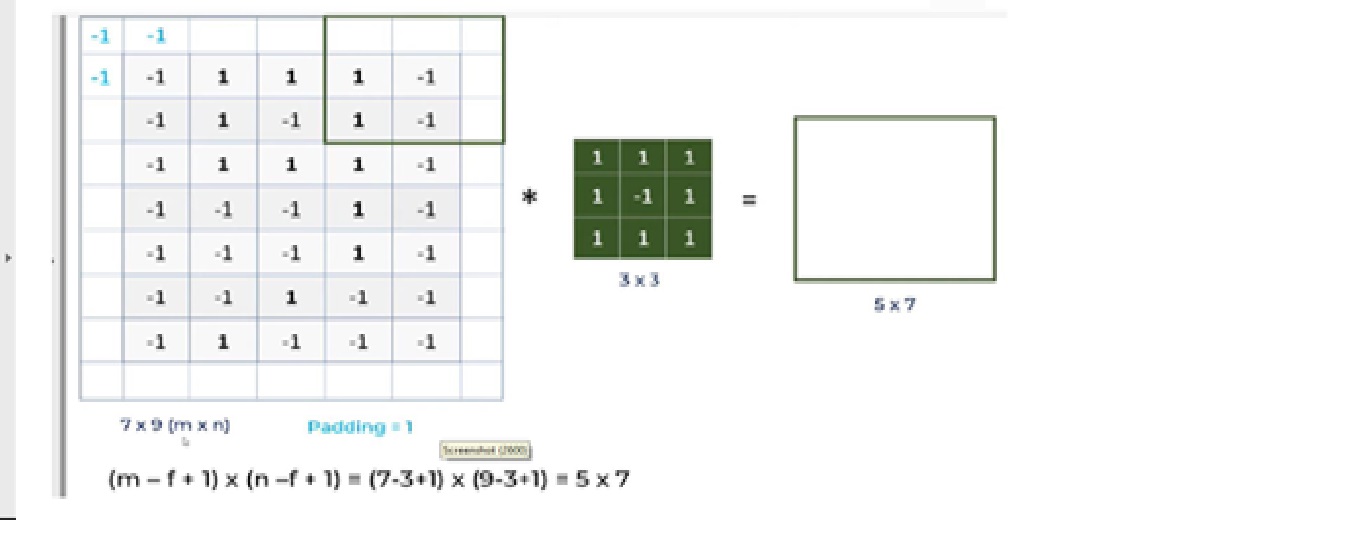
Feature detector which is known as a filter also is initialized randomly and then after a lot of iteration, filter matrix parameter selected which will be best for separating images. For instance, animals’ eye, nose, etc. will be considered as a feature which is used for classifying images using filter or feature detectors. Here we are using 16 features.

### Kernel\_size

Kernel\_size refers to filter matrix size. Here we are using a 2\*2 filter size.

### Padding

### FIG 5



Let’s discuss what is problem with CNN and how the padding operation will solve the problem.

a. For a gray scale (n x n) image and (f x f) filter/kernel, the dimensions of the image resulting from a convolution operation is (n – f + 1) x (n – f + 1).

So for instances, a 5\*7 image and 3\*3 filter kernel size, the output result after convolution operation would be a size of 3\*5. Thus, the image shrinks every time after the convolutional operation

b. Pixels, located on corners are contributed very little compared to middle pixels.

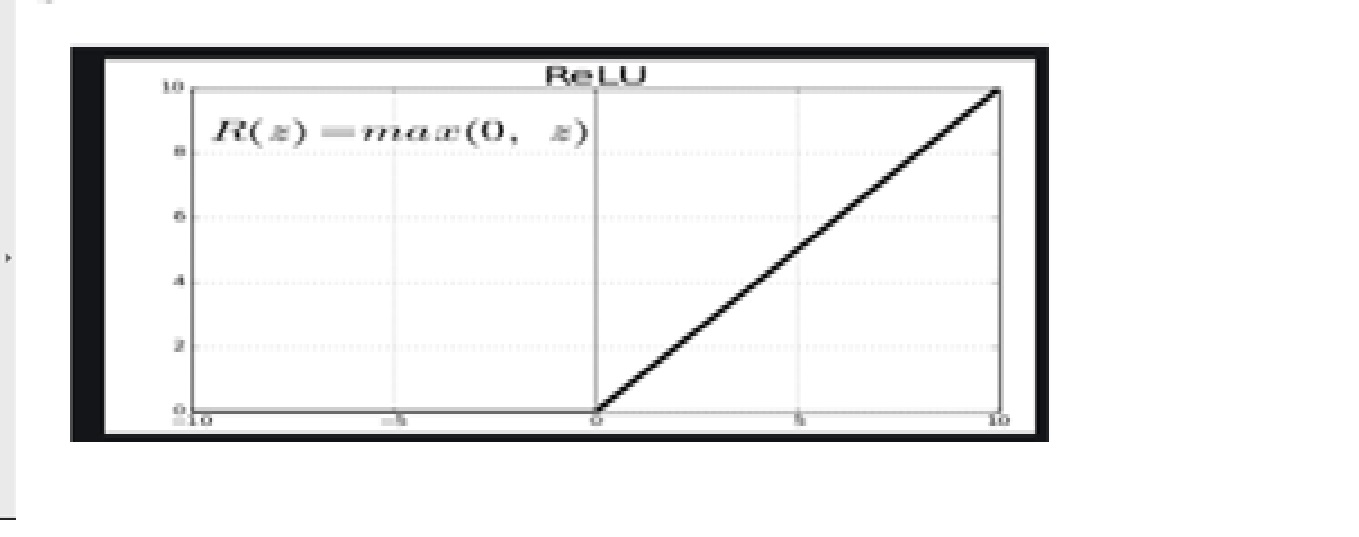
So, then to mitigate these problems, padding operation is done. Padding is a simple process of adding layers with 0 or -1 to input images so to avoid above mentioned problems.

Here we are using Padding = Same arguments, which depicts that output images have the same dimensions as input images.

### ·

### Activation Function – Relu

### FIG 6



Since images are non-linear, to bring non-linearity, the relu activation function is applied after the convolutional operation.

Relu stands for Rectified linear activation function. Relu function will output the input directly if it is positive, otherwise, it will output zero.

### Input shape

This argument shows image size – 224\*224\*3. Since the images in RGB format so, the third dimension of the image is 3.

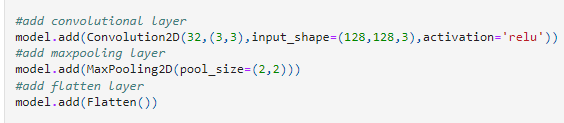
### Pooling Operation

Python Code :

model.add(MaxPooling2D(pool\_size=2))

We need to apply the pooling operation after initializing CNN. Pooling is an operation of down sampling of the image. The pooling layer is used to reduce the dimensions of the feature maps. Thus, the Pooling layer reduces the number of parameters to learn and reduces computation in the neural network.

FIG 7



**ADD DENSE LAYER**

Dense implements the operation: output = activation(dot(input, kernel) + bias) where activation is the element-wise activation function passed as the activation argument, kernel is a weights matrix created by the layer, and bias is a bias vector created by the layer (only applicable if use\_bias is True). These are all attributes of Dense.

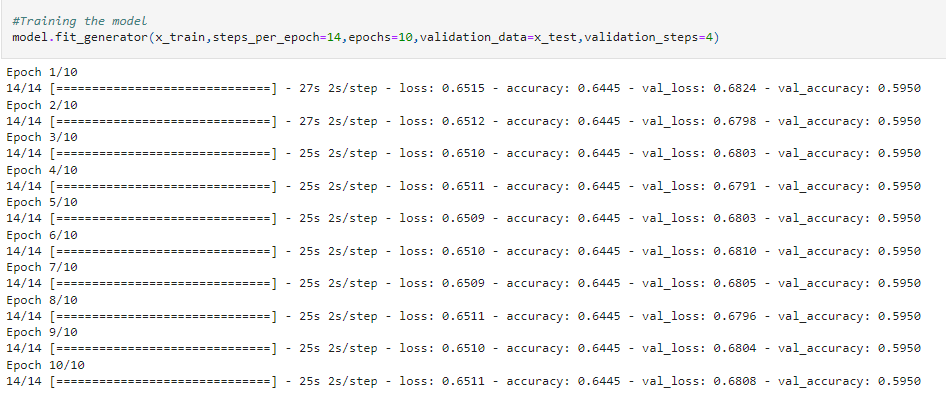
FIG 8

****

**TRAINING THE MODEL:**

he training step generates 3 datasets. 1) accuracy of the trained model, 2) the trained model, downloadable as a zip file, and 3) the trained model weights, downloadable as an hdf5 file. These files are needed for prediction in the next step.

FIG 9

****

**SAVE THE MODEL:**

Save your model by calling the save() function on the model and specifying the filename.

The example below demonstrates this by first fitting a model, evaluating it, and saving it to the file model.h5.

FIG 10

****

**PREDICTION OF THE MODEL:**

There are the following six steps to determine what object does the image contains?

* Load an image.
* Resize it to a predefined size such as 224 x 224 pixels.
* Scale the value of the pixels to the range [0, 255].
* Select a pre-trained model.
* Run the pre-trained model.
* Display the results.

FIG 11

****

CHAPTER 5

VIDEO ANALYSIS

**OpenCV for Video Processing**

OpenCV is an open-source library that provides us with the tools to perform almost any kind of image and video processing.

**Task 1: Capture Video from Camera**

Often, we have to capture the live stream with a camera. OpenCV provides a very simple interface to this. Let’s capture a video from the camera (I am using the in-built webcam of my laptop), convert it into grayscale video, and display it.

To capture a video, you need to create a **Video Capture** object. Its argument can be either the device index or the name of a video file. The device index is just the number to specify which camera. Normally one camera will be connected (as in my case). So I simply pass 0 (or -1). You can select the second camera by passing 1 and so on. After that, you can capture frame-by-frame. But in the end, don’t forget to release the capture. To read web cam will see the code.

OpenCV has a modular structure, which means that the package includes several shared or static libraries. The following modules are available:

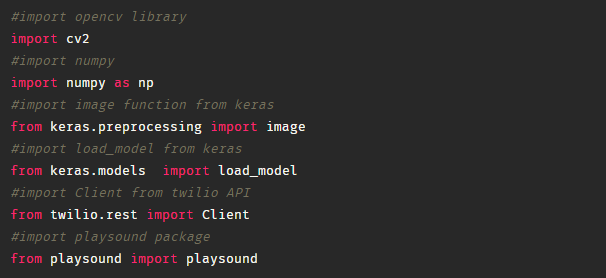
* [**Core functionality**](https://docs.opencv.org/4.x/d0/de1/group__core.html) (**core**) - a compact module defining basic data structures, including the dense multi-dimensional array Mat and basic functions used by all other modules.
* [**Image Processing**](https://docs.opencv.org/4.x/d7/dbd/group__imgproc.html) (**imgproc**) - an image processing module that includes linear and non-linear image filtering, geometrical image transformations (resize, affine and perspective warping, generic table-based remapping), color space conversion, histograms, and so on.
* [**Video Analysis**](https://docs.opencv.org/4.x/d7/de9/group__video.html) (**video**) - a video analysis module that includes motion estimation, background subtraction, and object tracking algorithms.
* [**Camera Calibration and 3D Reconstruction**](https://docs.opencv.org/4.x/d9/d0c/group__calib3d.html) (**calib3d**) - basic multiple-view geometry algorithms, single and stereo camera calibration, object pose estimation, stereo correspondence algorithms, and elements of 3D reconstruction.
* [**2D Features Framework**](https://docs.opencv.org/4.x/da/d9b/group__features2d.html) (**features2d**) - salient feature detectors, descriptors, and descriptor matchers.
* [**Object Detection**](https://docs.opencv.org/4.x/d5/d54/group__objdetect.html) (**objdetect**) - detection of objects and instances of the predefined classes (for example, faces, eyes, mugs, people, cars, and so on).
* [**High-level GUI**](https://docs.opencv.org/4.x/d7/dfc/group__highgui.html) (**highgui**) - an easy-to-use interface to simple UI capabilities.
* [**Video I/O**](https://docs.opencv.org/4.x/dd/de7/group__videoio.html) (**videoio**) - an easy-to-use interface to video capturing and video codecs.
* ... some other helper modules, such as FLANN and Google test wrappers, Python bindings, and others.

**Task 2: Importing the required libraries.**

Install Twilio library, run the below command in anaconda prompt,

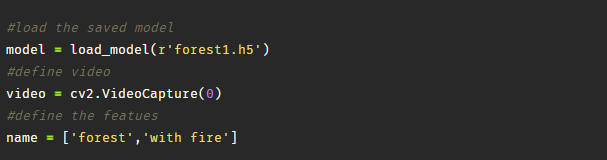
 “pip install twilio”.

FIG 12



**Task 3: Loading our saved model file using load\_model from Keras library**

FIG 13



**TWILIO SERVICE:**

Twilio is a customer engagement platform used by hundreds of thousands of businesses and more than ten million developers worldwide to build unique, personalized experiences for their customers. Twilio Engage is a new omnichannel marketing and growth platform that lets businesses of all sizes use the same kinds of tools, data integrations, analytics, and channels to build personalized campaigns that the digital giants use. Twilio Engage is the latest in the [suite of data management and analytics solutions](https://www.twilio.com/blog/introducing-twilio-engage-growth-platform) offered by Twilio Segment.

FIG 14



**SENDING ALERT MESSAGE:**

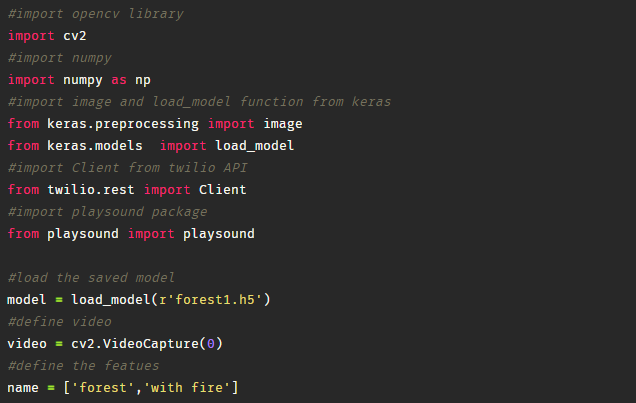
To play an alerting sound we need to install **"playsound"** library.

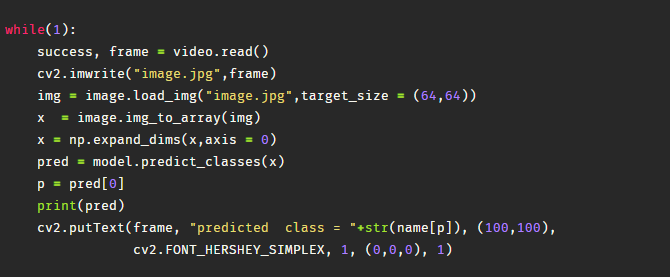
To install this library, open anaconda prompt and execute the below command.

Type**“pip install playsound”** click enter.

Combining all codes

FIG 15







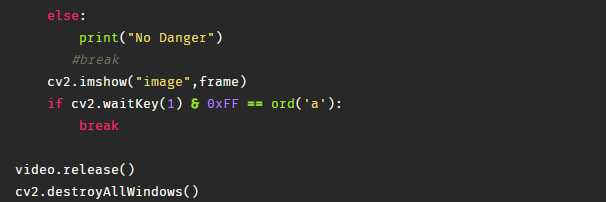
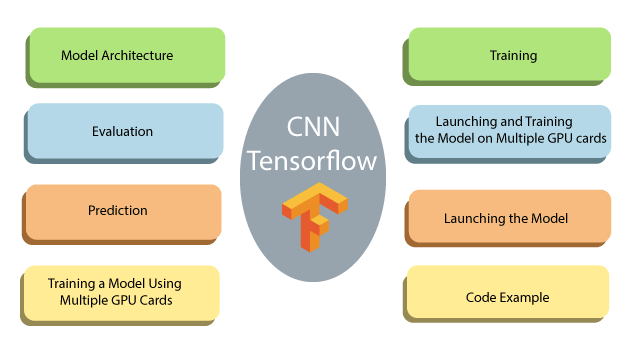


FIG 16

****

**CHAPTER 6**

**CONCLUSION**

In this model, the process of the forest fire image recognition algorithm based on CNN is presented. Its main feature is that the flame image is employed for training and testing. Then, AlexNet model is introduced, and an adaptive pooling method combined with color features is proposed for the problem that the traditional pooling method in CNN may weaken the image features in some cases. The effects of learning rate, batch size, and other parameters on the performance of CNN are analyzed based on experiments, and the optimal parameters are determined. Candidate flame area is extracted based on color feature; thus, the image feature of non-flame area in the hidden layer is reduced, and the feature, such as shape and texture, is enhanced. The information loss of image are avoided as adaptive pooling is adopted, and the rate of flame recognition in which fire area is segmentation than that of original image is adopted without segmentation. It is shown that the proposed algorithm has high recognition rate and is feasible. In this paper, the pooling of CNN is modified and applied on forest image recognition, recognition rate and consuming time will be developed deeply and compared with other algorithms in future.

CHAPTER 7

REFERENCES

[1] Official webpage of the European Forest Fire Information System at: http://effis.jrc.ec.europa.eu/

[2] Official webpage of the Copernicus Earth Observation Programme at: http://www.copernicus.eu

[3] Forest Fires in Europe, Middle East and North Africa 2016, JRC Science for policy report, BN 978-92-79-71292-0, ISSN 1831-9424, doi:10.2760/17690, available at: http://effis.jrc.ec.europa.eu/media/cms\_page\_media/40/Forest\_fires\_i

n\_Europe\_Middle\_east\_and\_North\_Africa\_2016\_final\_pdf\_JZU7He

L.pdf

[4] The 2018 Attica wildfires Wikipedia webpage available at: https://en.wikipedia.org/wiki/2018\_Attica\_wildfires

[5]László Földi and Rajmund Kuti, Characteristics of Forest Fires and their Impact on the Environment, Academic and Applied Research in Military and Public Management Science

(AARMS), Vol. 15, No. 1, 2016, pp. 5-17, ISSN 2064-0021;

[6] Seng Chuan TAY, Wynne HSU, Kim Hwa LIM, “Spatial Data Mining: Clustering of Hot Spots and Pattern Recognition”, Proceedings. 2003 IEEE International Geoscience and Remote Sensing Symposium, Volume: 6, pp: 3685- 3687, 21-25 July 2003.

[7] Young Gi Byun, Yong Huh, Kiyun Yu, Yong Il Kim, “Evaluation of Graph-based Analysis for Forest Fire Detections”, Proceedings of world academy of science, engineering and technology, volume. 10, December 2005, ISSN 1307-6884.

[8] Daniela Stojanova, Panče Panov, Andrej Kobler, Sašo Džeroski, Katerina Taškova, “Learning to predict forest fires with different Data Mining techniques”, Conference on Data Mining and Data Warehouses (SiKDD 2006),Ljubljana, Slovenia, pp. 255-258, October 9, 2006.

[9] Yasar Guneri Sahin, “Animals as Mobile Biological Sensors for Forest Fire Detection”, Sensors, vol. 7, pp. 3084-3099, 2007.

[10] Florent Lafarge, Xavier Descombes, Josiane Zerubia, “Forest fire detection based on Gaussian field analysis”, In Proc. European Signal Processing Conference (EUSIPCO), Poznan, Poland, September 2007.

[11] S. Movaghati, F. Samadzadegan, A. Azizi, “An agent-based algorithm for forest fire detection”, The International Archives of The Photogrammetry, Remote Sensing and Spatial Information Sciences, ISPRS Congress Beijing 2008, Volume. XXXVII, Part. B7, Commission: VII/4, Advanced classification techniques, 2008. ISSN 1682-1750.

[12] Adrian Ford and Alan Roberts, "Color Space Conversions", Technical report, August 11, 1998.

[13] CIE, Commission internationale de l'Eclairage proceedings, 1931. Cambridge University Press, Cambridge, 1932. [28] Smith, Thomas; Guild, John, "The C.I.E. colorimetric standards and their use", Transactions of the Optical Society, vol. 33, no. 3, pp. 73– 134, 1931-32.

[14] R. Duda and P. Hart, “Pattern Classification and Scene Analysis”, Bayes Decision Theory. John Wiley & Sons, pp. 10-13, 1973.

[15] P. Perona and J. Malik, “Scale space and edge detection using anisotropic diffusion", IEEE Transactions on Pattern Analysis and Machine Intelligence, Volume 12, Issue 7, July 1990, pp.629 - 639.

[16] L. Alvarez, P.L. Lions, and J.M. Morel, "Image Selective Smoothing and Edge Detection by Nonlinear Diffusion, II," SIAM Journal of Numerical Analysis, vol. 29, no. 3, pp. 845-866, 1992.

[17] Gerig, R. Kikinis, O. Kubler and F. A. Jolesz, “Nonlinear anisotropic filtering of MRI data”, IEEE Transactions on Medical Imaging , vol. 11, Issue 2 , pp. 221-232, June 1992.

[18] Sankar Seramani, Zhou Jiayin, Chan Kap Luk, N. Malmurugan and A. Nagappan, "Denoising of MR Images using Non Linear Anisotropic Diffusion Filtering as a Preprocessing Step", International Journal of BioSciences and Technology, Volume 1, Issue 1,2008, pp.17-21.

[19] G. Sapiro, “Cromaticity Diffusion”, Proc. of ICIP 2000, Vancouver, BC, Canada, Sept. 2000, vol. II, pp. 784-787.

[20] P. A. Maiellaro, R. Cozzolongo, P. Marino, “Artificial Neural Networks for the Prediction of Response to Interferon Plus Ribavirin Treatment in Patients with Chronic Hepatitis C”, Current Pharmaceutical Design, vol. 10, issue. 17, pp. 2101-2109(9), July 2004.

[21] P. Venkatesan and S. Anitha, "Application of a radial basis function neural network for diagnosis of diabetes mellitus", Current Science, vol. 91, no. 9, pp. 1195 - 1199, 10 November 2006.

[22] J.G. Wolff, “Medical Diagnosis as Pattern Recognition in a Framework of Information Compression by Multiple Alignment, Unification and Search”, Decision Support Systems, Volume 42, Issue 2, pp: 608 – 625, November 2006.

[23] Park, J. and Sandberg, I. W., “Approximation and radial basis function networks”, Neural Computation, vol. 5, pp. 305–316, 1993.

[24] Tou, J.T., Gonzalez, R. C., "Pattern Recognition Principles", Reading, MA: Addison – Wesley, 1974.

[25] Broomhead, D. S. and Lowe, D., “Multivariate functional interpolation and adaptive networks”, Complex Systems, vol. 2, pp. 321–355, 1988.