Detection of Human in Flames using HOG and SVM

A

Project Report

Submitted for the partial fulfilment Of B.Tech

Degree

in

INFORMATION TECHNOLOGY by

Sumit Kumar Gupta (1805232059)

Piyush Kumar Singh

(1900520139002)

Sahil Chaudhary (1805213047)

Under the supervision of

Dr Upendra Kumar

Ms Mudita Sharan

Department of Computer Science and Engineering

Institute of Engineering and Technology

Dr A.P.J. Abdul Kalam Technical University, Lucknow, Uttar Pradesh

June, 2022

CONTENTS

DECLARATION

We hereby declare that this submission is our own work and that, to the best of our belief and knowledge, it contains no material previously published or written by another person or material which to a substantial error has been accepted for the award of any degree or diploma of university or other institutes of higher learning, except where the acknowledgement has been made in the text. The project has not been submitted by us at any other institute for the requirement of any other degree.

Submitted by: - Date: 21/05/2022

(1) Name: Sumit Kumar Gupta
Roll No.: 1805232059
Branch: Information Technology

(2) Name: Piyush Kumar Singh
Roll No.: 1900520139002
Branch: Information Technology

(3) Name: Sahil Chaudhary
Roll No.: 1805213047
Branch: Information Technology

CERTIFICATE

This is to certify that the project report entitled "Detection of Human in Flames using HOG and SVM" presented by Sumit Kumar Gupta, Piyush Kumar Singh and Sahil Chaudhary in the partial fulfilment for the award of Bachelor of Technology in Infomation Technology, is a record of work carried out by them under my supervision and guidance at the Department of Information Technology at Institute of Engineering and Technology, Lucknow. It is also certified that this project has not been submitted to any other institute for the award of any other degrees to the best of my knowledge.

Dr Upendra Kumar
Ms Mudita Sharan
Department of Computer Science and Engineering
Institute of Engineering and Technology, Lucknow

ACKNOWLEDGEMENT

We would like to express our sincere thanks of gratitude to our supervisors **Dr Upendra Kumar** and **Ms Mudita Sharan** who gave us this wonderful opportunity to work on the project titled "Detection of Human in Flames using HOG and SVM". Our supervisora have helped us a lot in completing this project. We would like to thank our supervisors for their enthusiasm, patience, insightful comments, helpful information, practical advice and unceasing ideas that have helped us tremendously at all times during the completion of this project. We came to know about so many new things while working on this project.

Secondly, we would like to thank our family and friends who helped us a lot in finalizing the project within the limited time frame. We also express our sincere gratitude to the Department of Information Technology of the Institute of Engineering and Technology, Lucknow for supporting our project idea.

Thanks for all your encouragement!

Group Members:

1. Sumit Gupta
2. Piyush Singh
3. Sahil Chaudhary

ABSTRACT

The project helps detect the humans who are stuck in fire. The project can be broken down into two merging segments - the first is detecting the Fire, and the second is detecting the human. The fire detection algorithm detects the Fire and flames in the environment if present; the module works based on colour format YCbCr. It uses a Histogram of Oriented Gradient(HOG) and Support Vector Machine(SVM) to detect a human in the Fire. It evaluates several motion-based feature sets for human detection in the form of videos.

We integrated both the modules to make them work together. The method we used for detecting humans is a learning algorithm that uses a trained model containing various features of humans; for the detection of Fire. Four different rules involve colour thresholding and background differencing for moving object detection. Both algorithms are integrated by a condition that if the Fire is detected, then humans are to be detected using the algorithm. The main idea behind this technology is to spot the humans in the flames who are trapped in it so they can quickly be rescued. This can help the Firefighters in rapid planning and serious zone detection.

LIST OF FIGURES

LIST OF TABLES

Chapter 1

INTRODUCTION

1.1 Project background:

The current trends of research draw attention to two research areas that are combined and try to develop new technology for saving lives caused by the fire. The evolution in technologies in the field of computer vision. Szeliski, R [1] led to many new advancements in digital image processing. Schalkoff, R. J. [2] gave an idea to supplant the traditional and typical fire detection alarms with computer vision-based systems using image processing. It comprises three basic steps: the classification of the fire pixel, segmentation of moving objects, and then analyzing candidates region as the algorithm is based on image processing then there is less chance of raising false alarms. To distinguish moving pixels from non-moving pixels, a backdrop subtraction breakthrough with a frame differencing algorithm is applied to the frame buffer filled with successive frames of the input video. To raise a fire alarm, the moving pixels that are also detected as fire pixels are further studied in subsequent frames.

It is difficult to detect humans due to the differences in outfits, motion background, illumination, clutter, noise, and other factors that add to the complexity of human detection. We had algorithms like Viola-Jones, Viola, Jones, and others [4] for human identification, which were accurate enough to detect humans but had a few limitations that did not cause failure in real- time operation; according to Dalal, Triggs, and others [3]. Later, a new approach called Histogram of Oriented Gradient was developed to extract features from local cells. It is unaffected by geometric and photometric changes (HOG) except for object orientation. N. Dalal and B. Triggs [5]. On the other hand, Viola-Jones is reliant on geometric modification yet produces better results and is commonly used for facial recognition and detection. Al-Dabbagh, Barnouti, and others [6] By applying this technology in Autobots and surveillance to save lives and avoid fatalities, the technology may be advanced and used much more efficiently.

1.3 Existing Systems:

- It is a challenge to interpret that that the results are correct.
- High error-susceptibility- considerable amount of time is taken to identify the source and longer to correct one
- As Machine Learning models are open for Improvement but that depends upon Time and resources available, this proved to be a sturdy barrier to the accuracy of the model.

1.4 Proposed System:

- Fast and Efficient functioning- use of HOG removed the bottleneck for the model now it is invariant to geometric and photometric transformations, except for object orientation.
- Increased accuracy by providing an adequate amount of training data in both scenarios one with humans and the other where there were no humans.
- Efficient Algorithms like the sliding window, are used for feature comparison and other operations and this overall reduces the complexity of the system.

1.5 Project Scope:

Due to unexpected failures in the intended deliverables of the present fire security management system, our idea of detecting humans in fire using modern machine learning algorithms forms its shape, the project aims to serve a great purpose of saving lives, although currently, the project has some limited boundaries it has a great future scope ahead if it gets incorporated with modern tools like robots and drones, Our final project is a result of a complete cycle of collecting data, training a model and testing over multiples examples to achieve better efficiency at each step.

Chapter 2

LITERATURE REVIEW

2.1 Introduction:

Fires are posing an increasingly severe threat to people's lives and property. In the fire preprocessing stage, we combined the motion and color detection of the flame. In screening the fire candidate pixels, this strategy saves a significant amount of computing time. Second, despite its irregularity, the flame has a significant resemblance to the image's sequence. Many fire detection sensors are available nowadays to quickly detect fire and trigger fire alarms. Still, they can also generate false alarms in cigarette smoke due to chemical particles in the air that the sensors pick up. Second, despite its irregularity, the flame has a significant resemblance to the image's sequence. Many fire detection sensors detect fire and trigger fire alarms quickly. Still, they can also generate false alarms in cigarette smoke due to chemical particles in the air that the sensors pick up. To solve this problem, we employed computer vision and image processing to identify fire using the YCbCr color standard and to recognize humans in real-time using the Histogram of Oriented Gradient (HOG) and Support Vector Machine (SVM) classifier. This strategy, however, can be improved by incorporating this technology into a robot that can rescue persons trapped in a fire and save them from destruction.

That is to say, it fulfils two roles, detection of fire more precisely than the conventional systems available and majorly focusing on rescuing human beings stuck in a fire, which is very common to observe. Experimental results show that it could improve the accuracy and reduce the false alarm rate when comparedto conventional method.

2.2 Previous Works:

Also, prior work has been done in this field, which includes:

- **Khan et al.[7]** proposed a method based on video using flame dynamics and static indoor flame detection, using the colour, perimeter, area, and roundness of the flame. Their method takes a small fire such as a candle as an unimportant part. By removing and then applying the flame growth characteristics to judge, this method may have a big problem in early fire warnings.

- **Burnett and Wing[8]** used a new low-cost camera that can reduce the interference of smoke on the flame and has excellent detection capabilities for RGB and HSV, but there are still some limitations in the popularity and application of this camera.

- **Seebamrungsat et al.[9]** proposed a rule based on the combination of HSV and YCbCr. Their system requires extra conversion of colour space and is, therefore, better than using only one colour space method, but their work only uses the static characteristics of the flame. The method is relatively fragile and not stable enough.

Chapter 3

METHODOLOGY

3.1 Fire Detection

The fire detection algorithm detects the fire and flames in the environment if present, the module works based on colour format YCbCr, Celik and Demirel[10]. This colour space is chosen intentionally because of its ability to separate illumination information from chrominance more effectively than the other colour spaces. The rules defined for RGB colour space in order to detect possible fire-pixel or smoke-pixel candidates can be transformed into YCbCr colour space and analysis can be performed. However, the rules fall short in coming up with a single quantitative measure which can indicate how likely a given pixel is a fire pixel. We have observed that the fire samples show some deterministic characteristics in their colour channels of Y, Cb, and Cr which are shown below:

Fig 3.1(a)

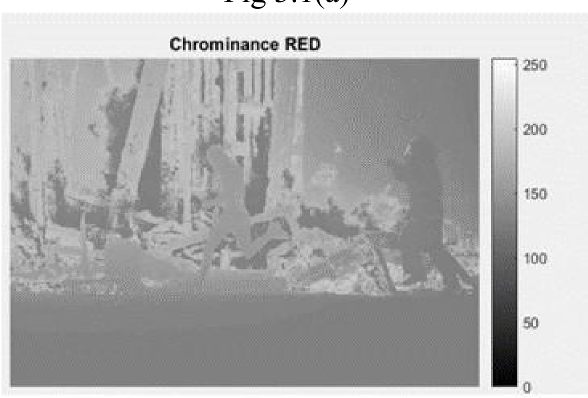


Fig 3.1(b)

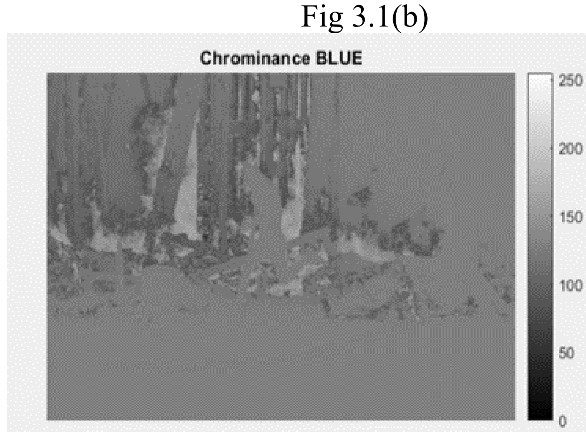


Fig 3.2 (c)

Fig 3.2 (d)

The above-given figures depict how the different components of YCbCr are extracted from the RGB channel of the original image. Sometimes this colour format is confused with the YUV colour format but the main difference is the YUV colour is analogue and YCbCr is digital.

The visualization of YCbCr is as follows:

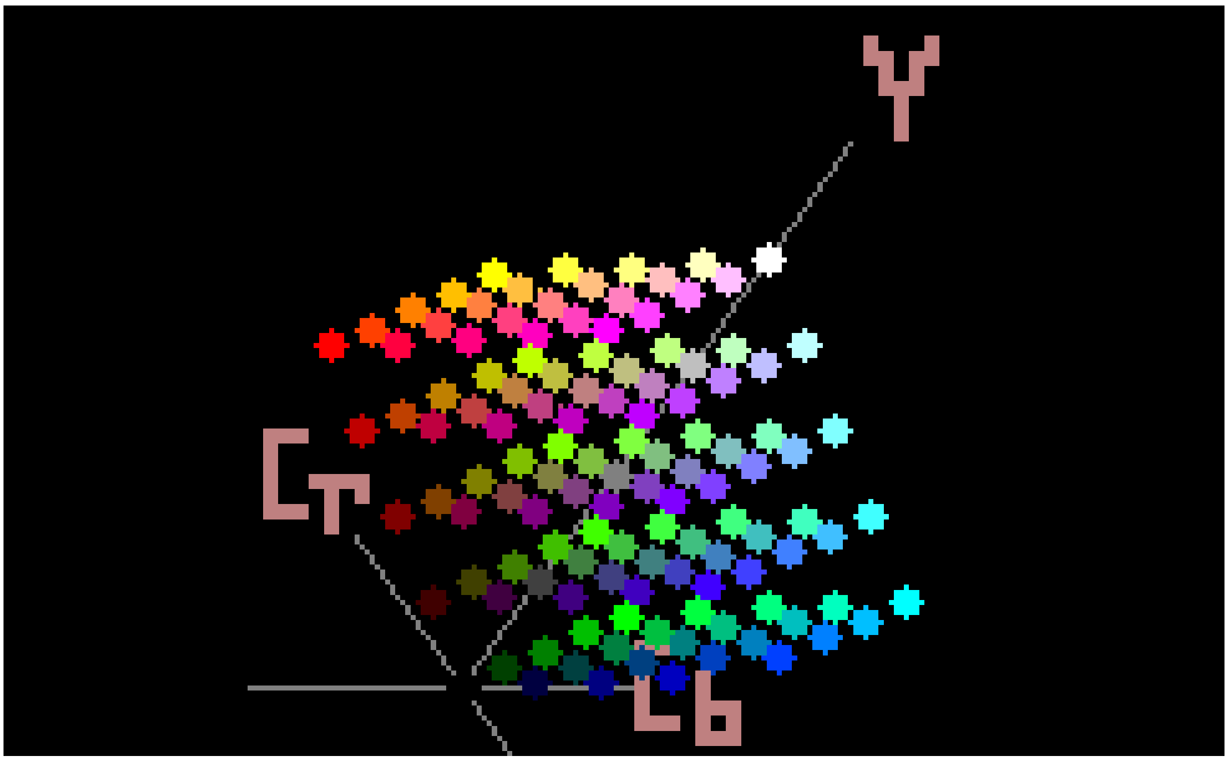


Fig 3.2

The YCbCr color space for fire can be obtained by the following method:

Y = 16 + R * 65.481 + G * 128.553 + B * 24.996; (i)

Cb = 128 + R* -37.797 - G * 74.203 + B * 112.0;. (ii)

Cr = 128 + R * 112.00 + G * -93.7864 + B * -18.214; (iii)

In the YCbCr colour space, Y′ is the luma component and Cb and Cr are the blue-difference and red-difference chrominance components and will be chosen according to the need to separate the illumination and detect the fire pixel. In YCbCr colour space for a fire pixel the satisfy the following condition

Y(x, y) >= Cr(x, y) >= Cb(x, y)

But the non-fire pixel doesn't satisfy the following condition, here (x, y) are the spatial location of fire pixel. The algorithm follows the rules given in order to detect the fire pixel:

Rule1: R1(x, y) = 1,

if ((R(x, y) > G(x, y)) && (G(x, y) > B(x, y)))

0, otherwise

Rule2: R2(x, y) = 1,

if (R(x, y) > 190) && (G(x, y) >100) && (B(x, y) < 140)

0, otherwise

Rule3: R3(x, y) = 1,

if Y(x, y) >= Cb(x,

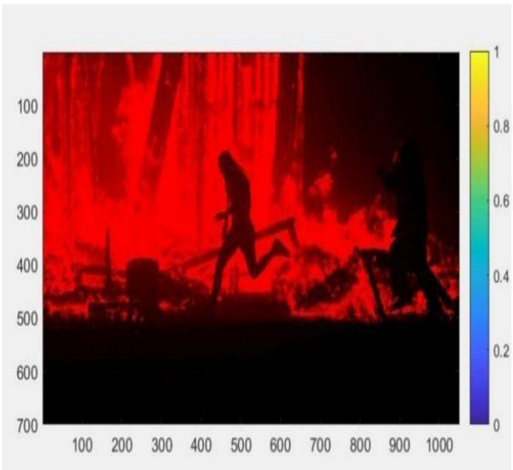
y) 0, otherwise

Rule 4: R4(x,y) = 1 ,

if (Cr(x, y) >= Cb(x, y)

0, otherwise

The outputs observed from the above set of rules are as displayed on the next page:



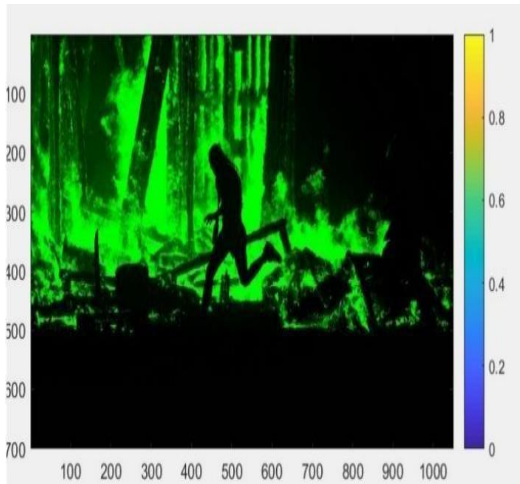
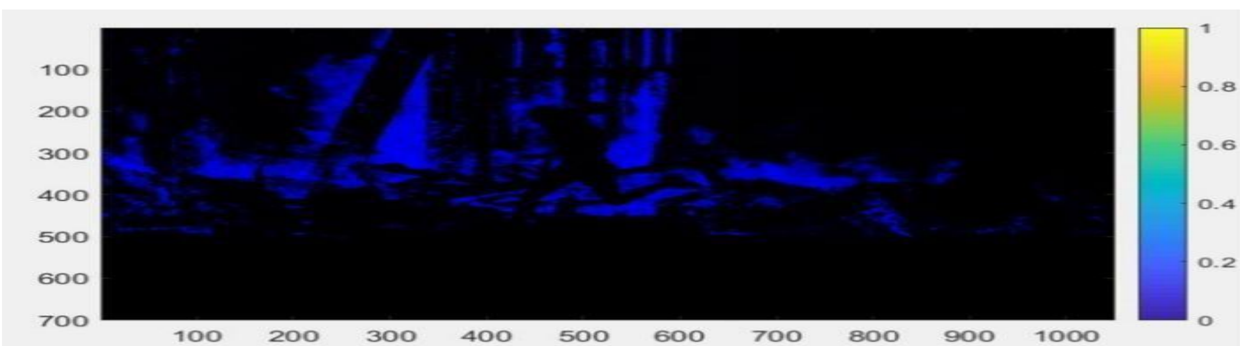
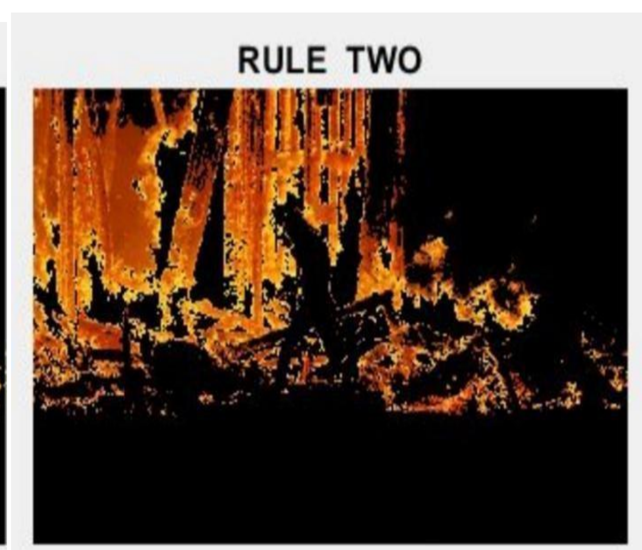


Fig 3.3(a)

Fig 3.3(b)

Fig 3.3(c)





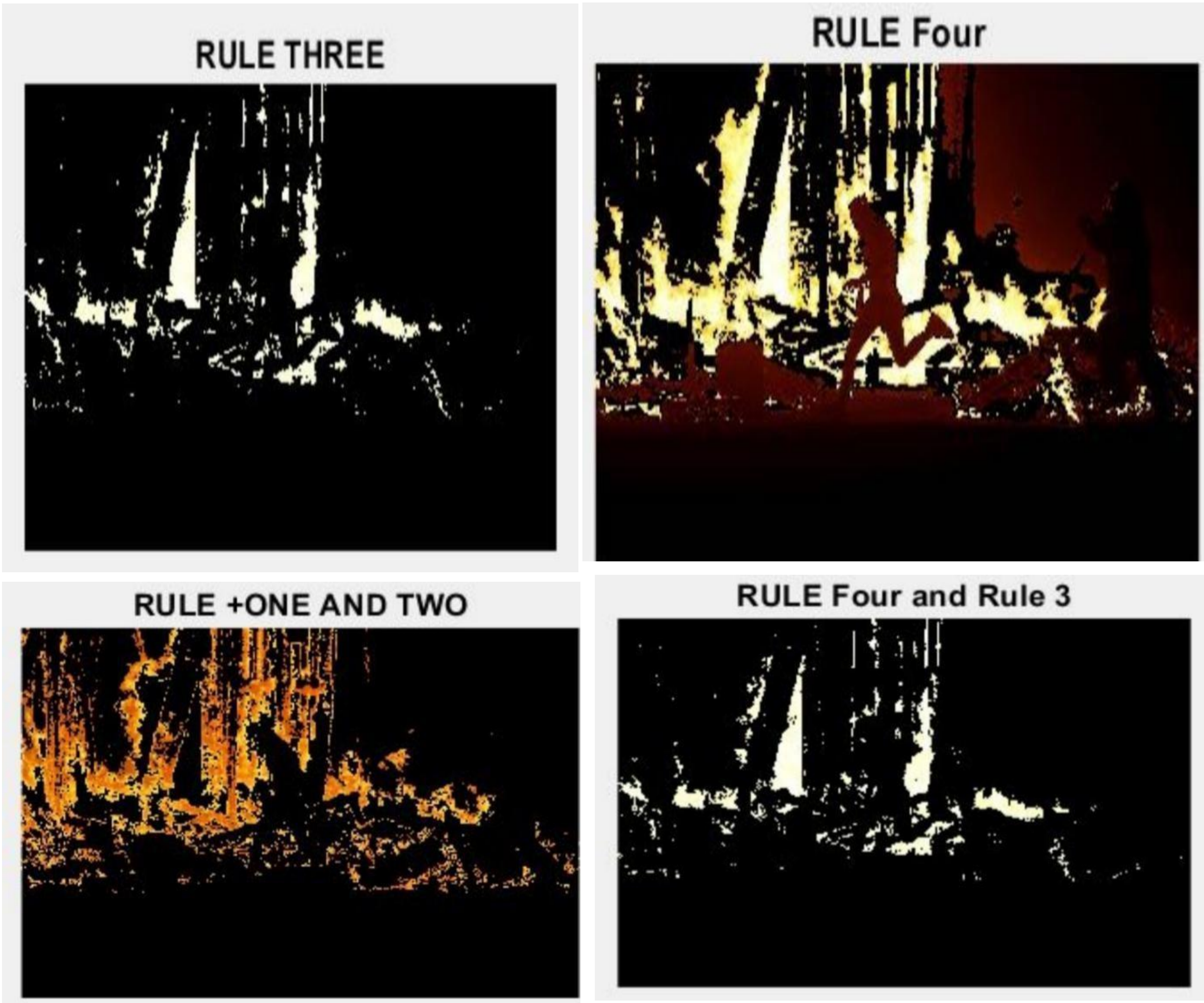
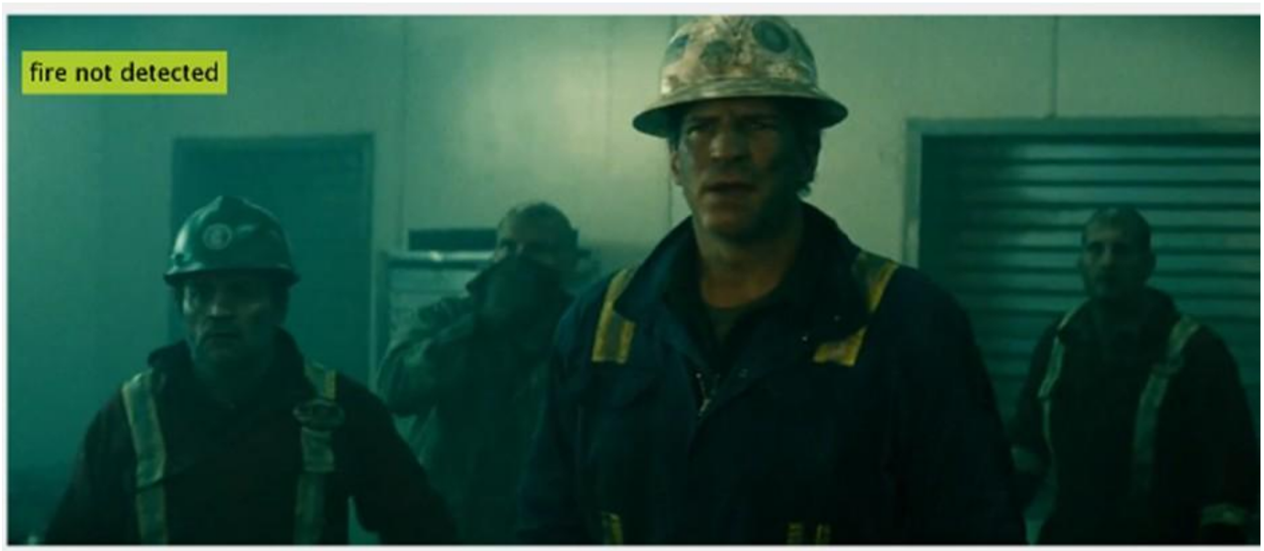


Fig 3.4

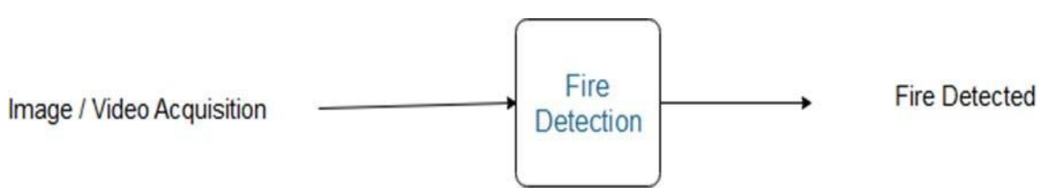
After the following results, we observe that the fire in the original image is detected

Results when a video was given as input:

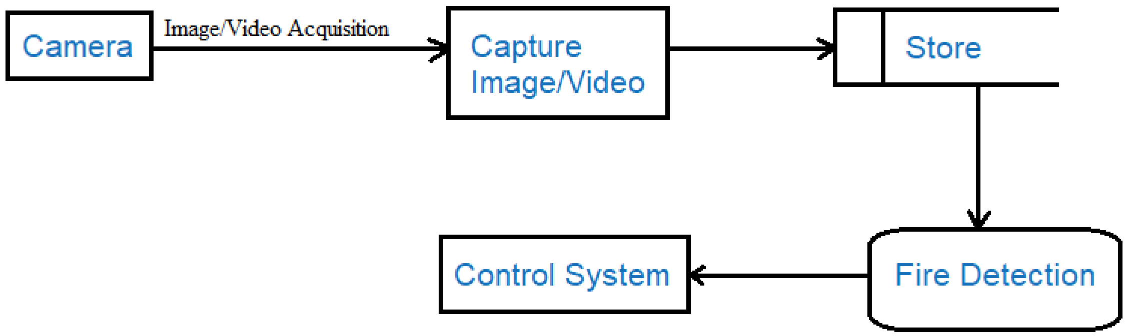


The data flow diagram of the following model is in this type:

Level Zero DFD of Fire Detection



Level one DFD of Fire Detection



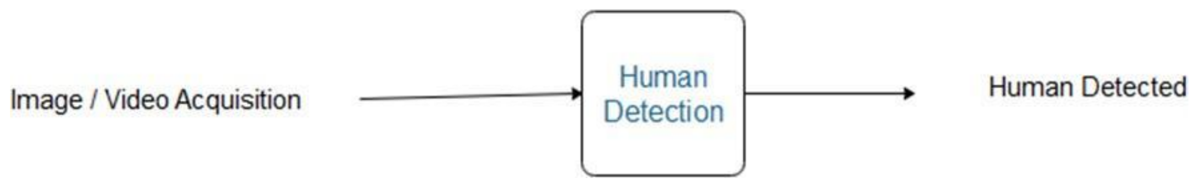
The critical task in a video sequencing is identifying the moving object which is an essential part of the object detection and for that, we have applied frame differencing which can be done by comparing two frames of the same time t, for instance, we take one frame from time instance t-1 and other from t-2 of the single time clock, as we know that the frames generated is 24 frames per second and those can be used to detect the moving objects in the video this method is very adaptive to other methods this procedure is further followed by noise removal to clarify the image we have used Gaussian filter for this purpose. Once the moving object is identified we detect the fire pixel, Celik, Ozkaramanlt and others [11] and the area by comparing two sequential frames generated and then check the maximum dispersion in the respective coordinates of X and Y and predict whether the fire is detected or not as shown.

3.2 Human detection

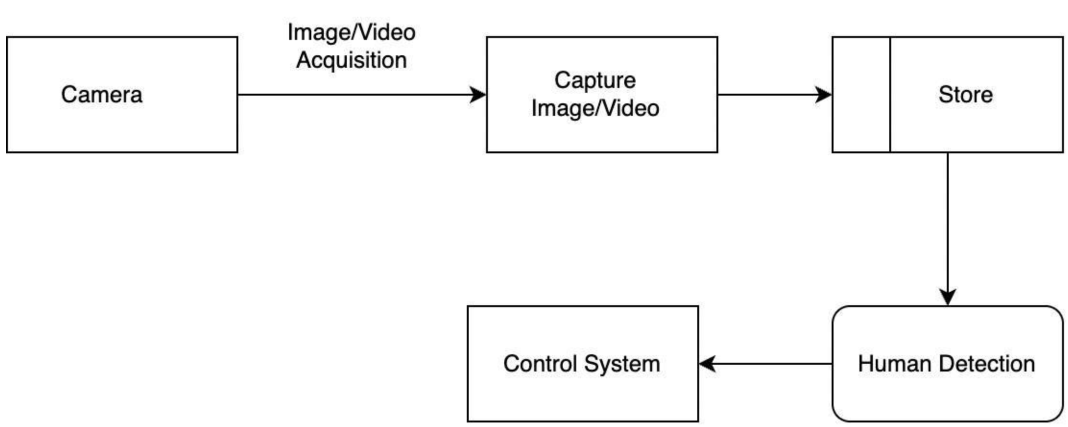
This paper introduces and evaluates a number of motion-based feature sets for human detection in videos. In particular, it studies oriented histograms of various kinds of local differences or differentials of optical flow as motion features, evaluating these both independently and in combination with the Histogram of Oriented Gradient (HOG) appearance descriptors that we originally developed for human detection in static images Papageorgiou, Oren and other [12]. The feature selection is a difficult task for human detection to differentiate humans from other objects as many researchers use body and silhouettes to differentiate from other objects but a human has different kinds of poses, gestures, articulation etc. which make it harder for any normal classifier to identify human surrounded by different kinds of objects.

To overcome this type of situation we have used a Support Vector Machine (SVM) as a classifier and pattern recognition. The working model of human detection in DFD form is as below:

Level Zero DFD of Human Detection



Level One DFD of Human Detection



The output of the following algorithm is as follows

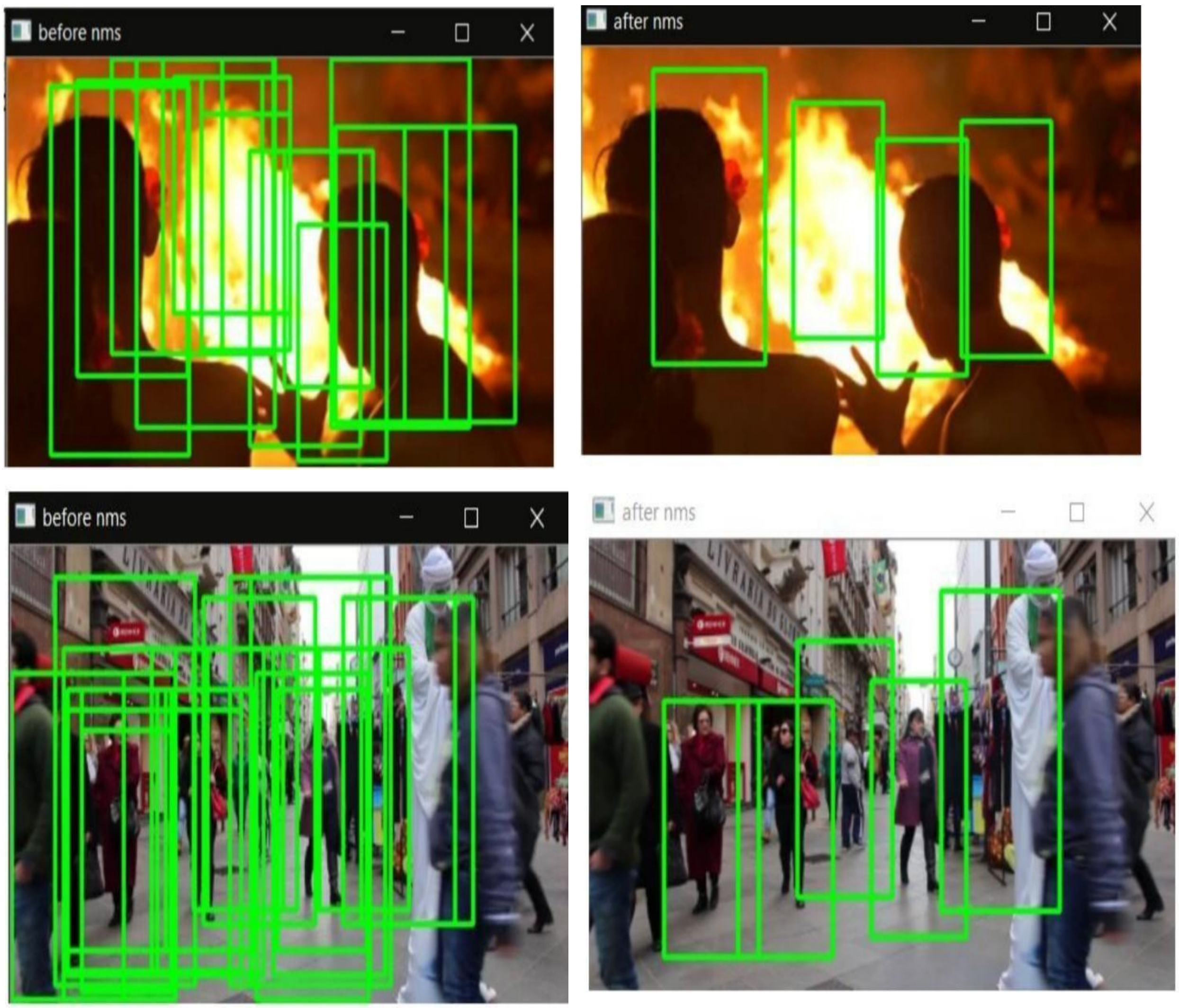


Fig. 3.6

The method uses dense grids of histograms of oriented gradients (HOG) as this method is powerful enough to classify humans and other objects. This method is generally time taking and to speed up this algorithm, Qiang Zhu, and Shai Avidan [13] proposed a fast human detection algorithm with variable blocks Xiaoyu Wang [14] used histograms of oriented gradients (HOG) and Local Binary Pattern (LBP) as the feature set, the method includes a global detector for whole scanning windows and part detector for local regions, the detection result is good. Colour information is used to detect humans. Sebastian Montabone and Alvaro Soto[15] present a human detection system based on visual saliency mechanism and color features. Here we simply used Histogram of Oriented Gradients to detect humans however histograms of oriented gradients (HOG) and Local Binary Pattern(LBP) can be considered in future in order to speed up the technology.

The Histogram of Oriented Gradients works in this way,

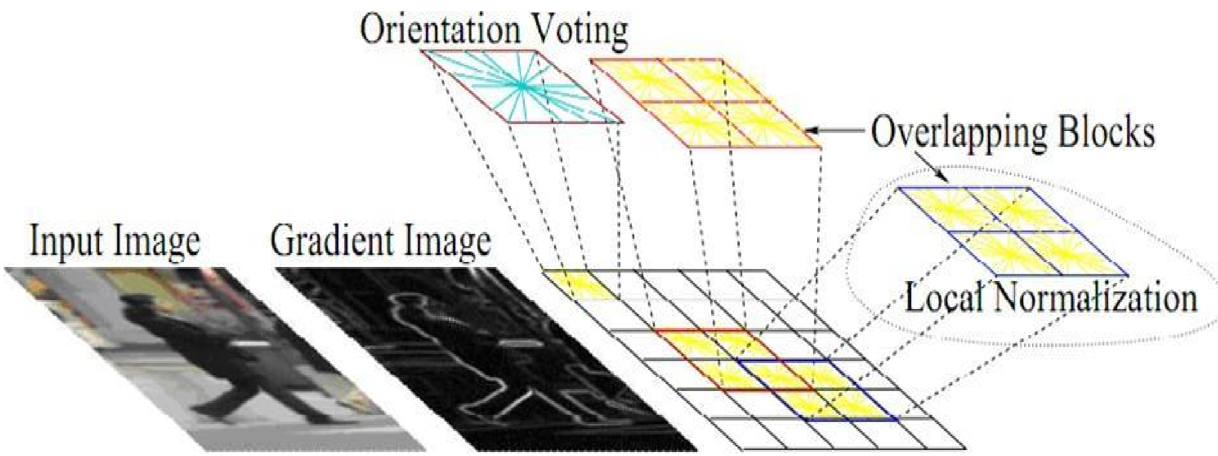


Fig 3.7

3.3 Proposed work

This paper mainly contributes to the two research areas in order to develop new technology that can be used in different fields to avoid destruction caused by fire. The technology can not only be used to save humans but can also be used to save the different objects as well by making slight changes in the module and the classifier.

Here we are working on two different modules which are capable of detecting humans in the fire if present and will be distinguishing the fire also from other objects present. We integrated both the modules to make them work together. Sometimes, fire becomes reason for great destruction causing life loss, loss of assets and many more. Keeping this in mind we have tried to develop new technology which constitutes of two algorithms, one for detecting humans and the other for detecting fire.

The method we used for detecting humans is a learning algorithm which uses a trained model containing various features of humans. For detection of Fire, four different rules involving color thresholding and background differencing for moving object detection. Both algorithms are integrated by a condition that if the fire is detected then humans are to be detected using the algorithm. The main idea behind this technology is to spot the humans in the flames who are trapped in it so they can easily be rescued. This can help the Firefighters in rapid planning and serious zone detection.

3.3.1 Implementing fire detection algorithm:

In fire detection algorithm it comprises three steps

- YCbCr component extraction
- Motion detection
- Area detection

We extract the YCbCr component from the RGB channel from the input video/image which is further filtered by applying different filters here we have used a Gaussian filter to remove the noise from the image for more clarity. In the next step, we apply background subtraction by frame differencing in which two sequential frames are compared to detect the moving object and once any moving object is identified then we detect the fire pixel in that area. The flowchart of the following algorithm is illustrated after following pseudo code.

The pseudo code for the following algorithm is as follows:

```
START
INPUT= video
GET number of
 frames
 n=numFrames;
 LOOP through each
        frame Im1= frame
        COMPUTE rchannel of Im1
        COMPUTE gchannel of Im1
        COMPUTE bchannel of Im1
        COMPUTE Ydash = 16+(0.2567890625 * rchannel)+ (0.50412890625 * gchannel)
                                    +(0.09790625 * bchannel)
        COMPUTE Cb= 128+(-0.14822265625 * rchannel)-(0.2909921875 * gchannel)
                                +(0.43921484375* bchannel)
       COMPUTE Cr = 128+(0.43921484375 * rchannel)- (0.3677890625 * gchannel)
                               -(0.07142578125 *
       bchannel) COMPUTE Ymean=
       (mean(mean(Ydash))); COMPUTE Cbmean=
       (mean(mean(Cb))); COMPUTE Crmean=
       (mean(mean(Cr))); COMPUTE Crstd= std2(Cr);
```

#Rule 1

```
IF Ydash>Cb THEN
ASSIGN respective row and column indices to R1r and R1c
SET ruleIpixel AS length of R1r
SET Ir1 AS zero matrix
LOOP THROUGH ruleIpixel-1
  Ir1(R1r(i),R1c(i),1) =rchannel(R1r(i),R1c(i),1)
  Ir1(R1r(i),R1c(i),2) =gchannel(R1r(i),R1c(i),1)
  Ir1(R1r(i),R1c(i),3) =bchannel(R1r(i),R1c(i),1)
  i=i+1;
END LOOP
```

#Rule 2

```
IF Ydash >Ymean AND Cr > Crmean
ASSIGN respective row and column indices to R2r and R2c
END IF
SET ruleIIpixel AS length of
R1r SET Ir2 AS zero matrix
LOOP THROUGH ruleIIpixel-1
  Ir2(R2r(i),R2c(i),1) = rchannel(R2r(i),R2c(i),1)
  Ir2(R2r(i),R2c(i),2) = gchannel(R2r(i),R2c(i),1)
  Ir2(R2r(i),R2c(i),3) = bchannel(R2r(i),R2c(i),1)
  i=i+1;
END LOOP
```

#Rule one and two

```
IF Ydash>Cb AND Ydash>Ymean AND Cr > Crmean
THEN ASSIGN respective row and column indices to R12r and
R12c
END IF
SET ruleI_IIpixe AS length of
R12r SET Ir12 AS zero matrix
LOOP THROUGH ruleI_IIpixel-1
  Ir12(R12r(i),R12c(i),1) =rchannel(R12r(i3),R12c(i))
  Ir12(R12r(i),R12c(i),2) =gchannel(R12r(i3),R12c(i))
  Ir12(R12r(i),R12c(i),3) =bchannel(R12r(i3),R12c(i))
  i=i+1;
END LOOP
```

#Rule three

```
IF Ydash>Cr AND Cb>Cr THEN
ASSIGN respective row and column indices to R3r and R3c
END IF
SET ruleIIIpixel AS length of
R3r SET Ir3 AS zero matrix;
LOOP THROUGH ruleIIIpixel-1
Ir3(R3r(i),R3c(i),1) =rchannel(R3r(i),R3c(i))
Ir3(R3r(i),R3c(i),2) =gchannel(R3r(i),R3c(i))
Ir3(R3r(i),R3c(i),3) =bchannel(R3r(i),R3c(i))
i=i+1;
END LOOP
```

#Rule four

```
IF Ydash>Cr AND Cb>Cr THEN
ASSIGN respective row and column indices to R4r and R4c
END IF
SET ruleIVpixel AS length of
R4r SET Ir4 AS zero matrix
LOOP THROUGH ruleIVpixel-1
    Ir4(R4r(i),R4c(i),1) =rchannel(R4r(i),R4c(i))
    Ir4(R4r(i),R4c(i),2) =gchannel(R4r(i),R4c(i))
    Ir4(R4r(i),R4c(i),3) =bchannel(R4r(i),R4c(i))
    i=i+1;
END LOOP
```

#Rule three and four

```
IF Ydash > Cr AND Cb > Cr AND Cr < 7.4*Crstd THEN
    ASSIGN respective row and column indices to R6r and R6c
END IF
SET ruleVIpixel AS length of R6r
SET Ir6 AS zero matrix
LOOP THROUGH ruleVIpixel-1
    Ir6(R6r(i),R6c(i),1) =rchannel(R6r(i),R6c(i))
    Ir6(R6r(i),R6c(i),2) =gchannel(R6r(i),R6c(i))
    Ir6(R6r(i),R6c(i),3) =bchannel(R6r(i),R6c(i))
```

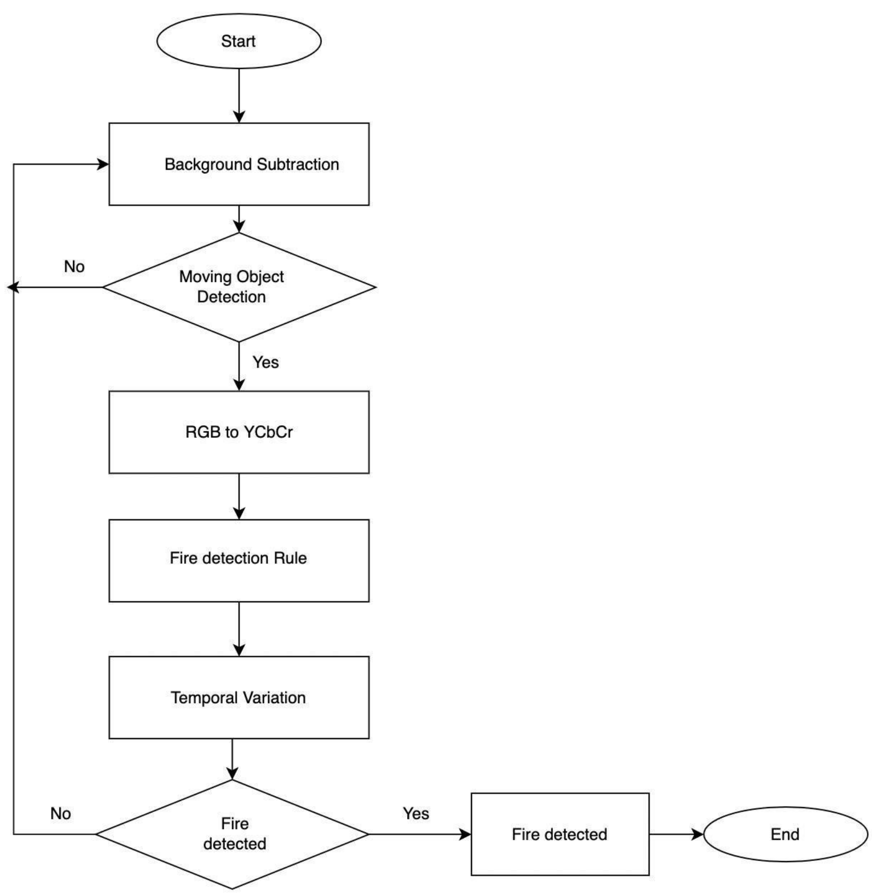
```
        i6=i6+1
    ;
    END LOOP

    ASSIGN Ir12+Ir6 TO f_f

        IF f_f(:,:,1)> f_f(:,:,3) AND f_f(:,:,2) > f_f(:,:,3) AND f_f(:,:,3 )< 100 THEN
       ASSIGN respective row and column indices to R7r and R7c
    END IF
    ASSIGN rows, columns dimension of (Im1 (:,:,1))
    ASSIGN ff_r,ff_c dimension of R7r
        IF    (ff_r>(    rows*columns*4/100)     )
        THEN    WRITE   'fire   detected'   ON
        current frame SHOW frame
     ELSE
        WRITE 'fire NOT detected' ON current
        frame SHOW frame
    END IF

END LOOP
END
```

The flowchart of the following algorithm is: (Fig 3.8)



3.3.2 Implementing human detection algorithm

In human detection algorithm undergoes the following steps:

- Motion detection
- HOG feature extraction
- Applying SVM classifier

The essential task of object detection is to detect the moving object which reduces the task of the sliding window for each pixel in the captured video or image so that it can work on the pixels identified as having differences. This task is done by taking two consequent frames and comparing them this process is the same as we did in the fire detection algorithm. Further, the HOG descriptor is applied to extract the feature which includes four stages.

The first stage is it applies a template to count the gradients of every point.

The second stage separates the images into small spatial sections, which we call cells. It then accumulates a local 1-D histogram of the oriented gradient for each cell in the overall image pixel, which we refer to as an oriented histogram representation.

The third stage is the normalization computation, which improves the image's illumination and contrast. This is accomplished by aggregating the energy of the local histogram over blocks (groups of cells are called blocks). The result is used to normalize each cell in a single block, referred to as a Histogram of Oriented Gradients.

The final stage compiles all HOG descriptors from the detection window's blocks into a feature vector for classification.

The pseudo code of feature extraction is as follows:

```
INPUT= img_pos      # positive sample images
INPUT= img_neg      # negative sample images

minimum_window_size = [64, 128] #sliding window size
step_size = [10, 10]
orientations = 9        #gradient (number of orientations in a histogram)
pixels_per_cell = [6,6]
cells_per_block = [2, 2]

CREATE directory POS_MOD #for positive feature storage
CREATE directory POS_NEG # for negative feature storage
```

For each positive sample image
READ img_pos AS grayscale

FUNCTION hog **#feature detection function**
PASS IN: img_pos, orientations, pixel_per_cell, cells_per_block, block_normalise, visualise=True, transform_sqrt=False, feature_vector

Square-root normalization

- $\mathbf{G_x=I*D_x, G_y=I*D_y}$ compute the image gradient in both the x and y-direction.
- $\mathbf{|G|=\sqrt{(G_Y^2+G_x^2)}}$ compute the final gradient magnitude representation of the image.
- $\mathbf{\Theta=arctan2(G_yG_x)}$ computes the orientation of the gradient for each pixel in the input image and divide image into cells and blocks.
- Construct a histogram of oriented gradients using gradient magnitude **|G|** and orientation **Θ** each pixel contributes a weighted vote to the histogram. #The weight of the vote is simply the gradient magnitude at the given pixel.
- Compute a histogram of oriented gradients using 9 orientations #for each cell.
- Apply block normalization.
- Concatenate the resulting histograms.

PASS IN: img_neg, orientations, pixel_per_cell, cells_per_block, block_normalise, visualize = True, transform_sqrt=False, feature_vector

Square-root normalization

- $G_x=I*D_x, G_y=I*D_y$ compute the image gradient in both the x and y-direction.
- $|G|=\sqrt{} G_X^2+G_x^2$ compute the final gradient magnitude representation of the image.
- $\Theta=arctan2(G_yG_x)$ computes the orientation of the gradient for each pixel in the input image and divide image into cells and blocks.

- Construct a histogram of oriented gradients using gradient magnitude **|G|** and orientation **Θ** each pixel contributes a weighted vote to the histogram. #The weight of the vote is simply the gradient magnitude at the given pixel.
- Compute a histogram of oriented gradients using 9 orientations #for each cell
- Apply block normalization
- Concatenate the resulting histograms

PASS OUT: A visualization of the HOG image AS fd #a real-valued feature vector for **negative** images.
END FUNCTION
SET filename
WRITE fd IN filename
STORE filename in POS_NEG
END FOR

The next step of the algorithm is to classify the object using a classifier here we are using Support Vector Machine (SVM), this classifier is widely used, it computes a high dimensional hyperplane to separate the categories of the different objects. To calculate the plane, the chosen image feature space or a kernel of this feature space is used.

The first stage is the training stage in which the model is trained using this classifier, the classification function C(x) is defined as:

$C(x)=a_i\, y_j\, .k(x_i, x)+b$

The second stage is the training stage we link the value with human detection. If C(x) >0, the sliding window can be regarded as a human region, the higher the C(x), the higher the a human region, the higher the C(x), the higher the confidence. On the contrary, if C(x) <0, the detection window can be regarded as other objects.

The pseudo-code for the SVM classifier is as shown:

```
START
INPUT= img_pos                # positive sample images
INPUT= img_neg                # negative sample images

SET minimum_window_size = [64, 128]          #sliding window size
SET step_size = [10, 10]
SET orientations = 9              #gradient (number of orientations in a histogram)
SET pixels_per_cell = [6, 6]
SET cells_per_block = [2, 2]
DECLARE ARRAY samples []
DECLARE ARRAY labels []
FOR each positive sample image
        READ img_pos
        FUNCTION resize
        PASS IN: img_pos, (320,240)
        Resizes the image
        PASS OUT: resimg
        END FUNCTION
        FUNCTION hog              #feature detection function
                PASS IN: resimg, orientations, pixel_per_cell, cells_per_block, block_normalise,
        visualise=True, transform_sqrt=False, feature_vector
        compute a histogram of oriented gradients
        PASS OUT: HOG AS hist
        END FUNCTION
        APPEND hist IN samples []
        APPEND 1 IN labels []
END FOR

FOR each negative sample image
        READ img_neg
        FUNCTION resize
        PASS IN: img_neg, (320,240)
        Resizes the image
        PASS OUT: resimg
        END FUNCTION
        FUNCTION hog               #feature detection function
                PASS IN: resimg, orientations, pixel_per_cell, cells_per_block, block_normalise,
        visualise=True, transform_sqrt=False, feature_vector
        compute a histogram of oriented gradients
        PASS OUT: HOG as hist
        END FUNCTION
        APPEND hist in samples []
        APPEND 0 in labels []
```

END FOR

SHUFFLE samples and labels

FUNCTION SVM.create
PASS IN: nothing
Creates empty model
PASS OUT: nothing
END FUNCTION

FUNCTION SVM.settype
PASS IN: ‘type’
sets the type of created empty model
PASS OUT: nothing
END FUNCTION

FUNCTION SVM.setKernel
PASS IN: ‘kerneltype’
sets the type of kernel
PASS OUT: nothing
END FUNCTION

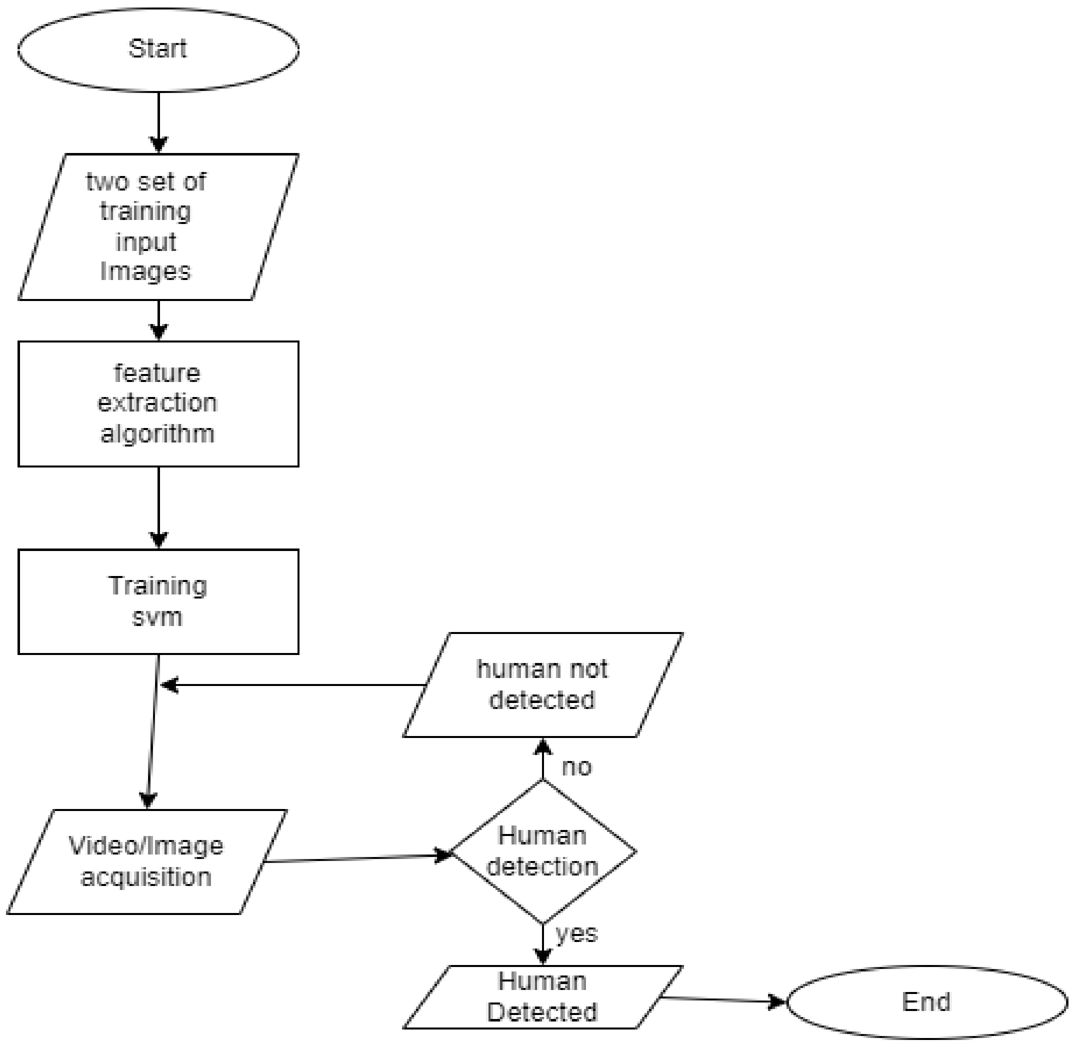
FUNCTION SVM.train
PASS IN: samples, labels
trains the model
PASS OUT: trained
model END FUNCTION

FUNCTION SVM.save
PASS IN: path to store model
(directory) saves the model
PASS OUT: nothing
END FUNCTION

END

The flowchart of human detection is as follows:

Fig 3.9



Chapter 4

EXPERIMENTAL RESULTS

The above analysis is tested under different conditions to determine the accuracy and performance of algorithms and assure the reliability of the technology. We have tested the fire detection algorithm over different images and found the positive results, the videos of all definitions were used to know its efficiency, various kinds of pictures were used with different backgrounds and was found that the algorithm works good enough in detecting the fire and the flames present in the picture. It can also be used to detect the forest fire but sometimes when the sun is either rising or setting means the colour intensity of the sun is close to the fire then it considers the sun also like fire, however, this problem can be overcome by using deep learning and neural networks as it will give the less false alarm of fire detection. When we gave low-quality video as an input to test the fire detection algorithm we observed that low-quality video takes less time to process and predict the output in comparison to high-quality video.

As for human detection, we used 1800 training images which included positive and negative images. Positive images contained humans whereas negative images had no humans, both shared the same background in order to train the model to avoid false detection the number of negative images is more than the positive ones. To validate the human detection algorithm we tested the algorithm on different videos and images and found out that the algorithm works well on high resolution(1920 x1080) videos only as the algorithm is time taking so for its better performance we can use HOG+LBP for faster and better results. The accuracy of the algorithm depends upon the training and the classifier.

Given below are the images of a running video, which has been demonstrated as demo as of how it actually works and the accuracy upto very acute details and as very clearly seen in figure 4.3, that it does not indicate smoke as fire. figure 4.4 gives the final output, that is the detection of fire .

Fig 4.1

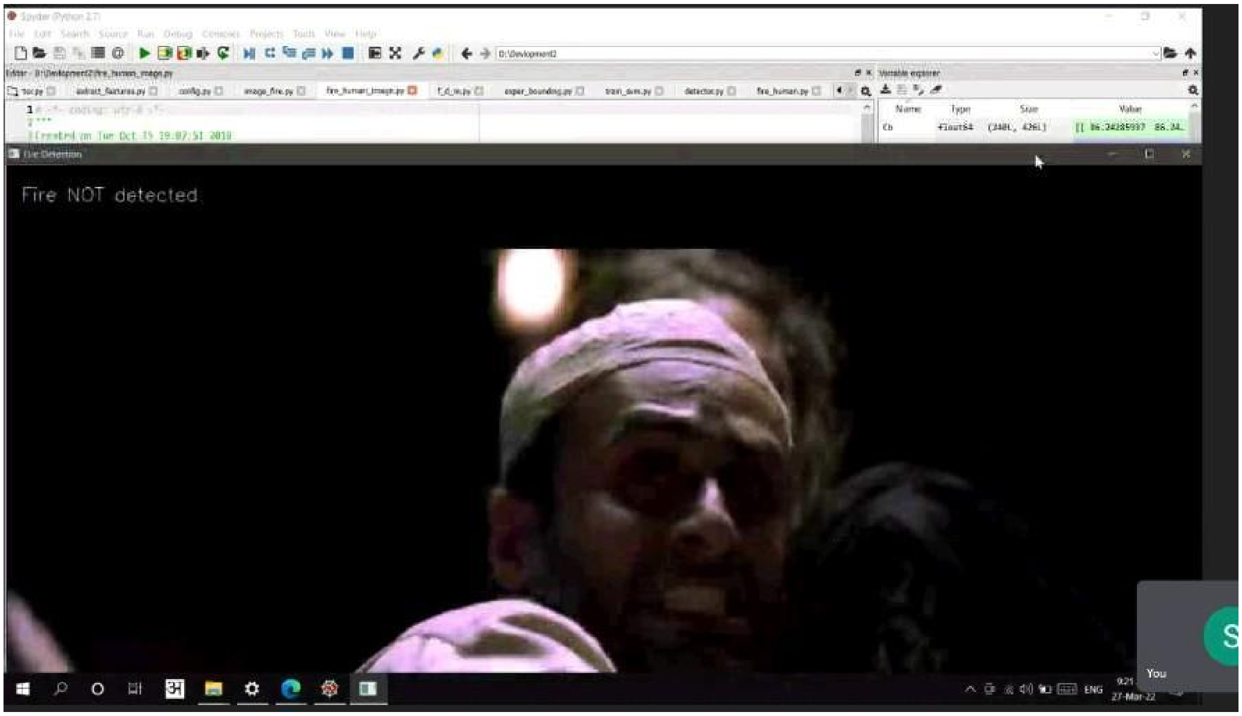


Fig 4.2

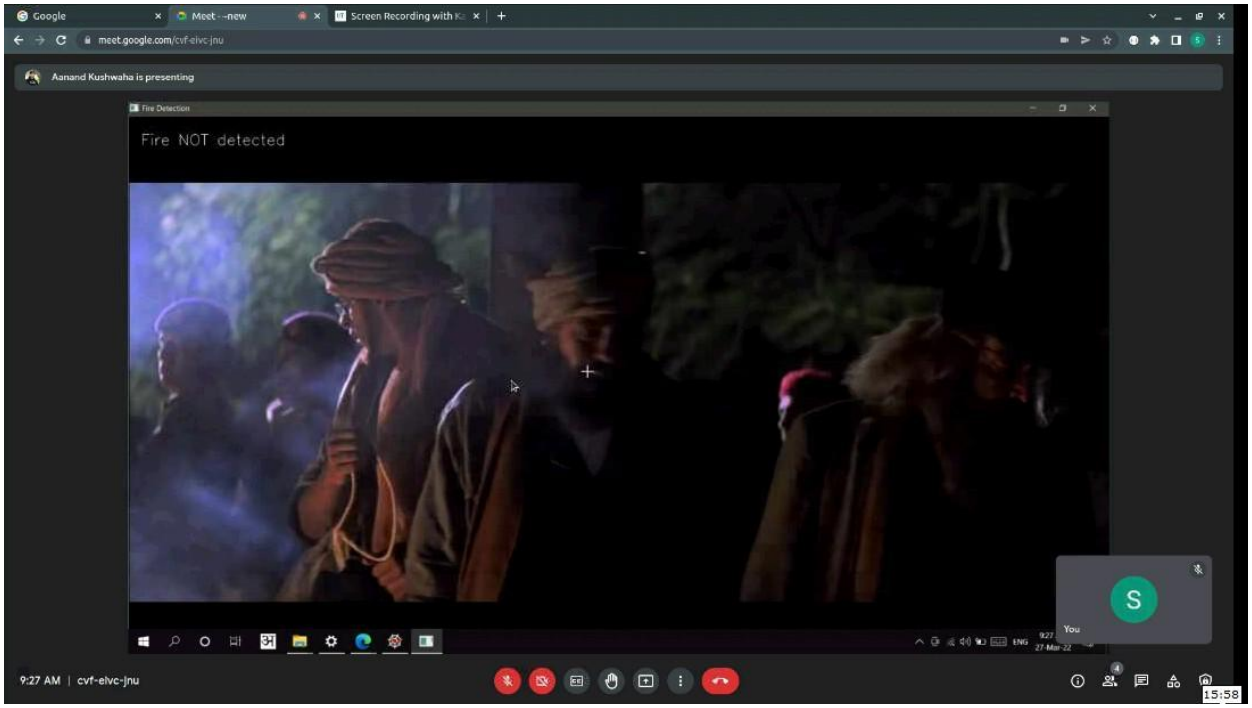

Google
Meet - -new
Screen Recording with K
meet.google.com/cvf-elvc-jnu
Aanand Kushwaha is presenting
Fire Detection
Fire NOT detected
You
9:27 AM | cvf-elvc-jnu
15:58

Fig 4.3


Aanand Kushwaha is presenting
Fire Detection
Fire detected
You
Turn on microphone (ctrl + d)
9:28 AM | cvf-elvc-jnu

Fig 4.4

Table 1.1 Experimental Table

Exp No	Actual	Predicted Human	+ve or -ve	Detected	Term
Exp 1	H	H	+ve	✅	True Positive
Exp 2	N	H	+ve	❌	False Positive
Exp 3	N	N	-ve	✅	True Negative
Exp 4	H	N	-ve	❌	False Negative
Exp 5	H	H	+ve	✅	True Positive
Exp 6	N	N	-ve	✅	True Negative
Exp 7	H	N	-ve	❌	False Negative
Exp 8	N	N	-ve	✅	True Negative
Exp 9	N	N	-ve	✅	True Negative
Exp 10	H	H	+ve	✅	True Positive
Exp 11	H	H	+ve	✅	True Positive
Exp 12	N	H	+ve	❌	False Positive
Exp 13	H	H	+ve	✅	True Positive
Exp 14	H	H	+ve	✅	True Positive
Exp 15	H	H	+ve	✅	True Positive
Exp 16	N	N	-ve	✅	True Negative
Exp 17	H	H	+ve	✅	True Positive
Exp 18	N	N	-ve	✅	True Negative
Exp 19	H	H	+ve	✅	True Positive
Exp 20	N	N	-ve	✅	True Negative

H- Human Presence, N - Absence of Human

True Positive - 810, True Negative - 630, False Positive - 180, False Negative - 180

Confusion Matrix -

Table 1.2 Confusion Matrix

	Predicted Negative	Predicted Positive
Actual Negative	630	180
Actual Positive	180	810

- Precision - What Percent of Positive prediction made Correct.

Precision = $True\ Positive/(True\ Positive + False\ Positive)$
= 810/(810+180)
= 9/11
= 0.81

- Recall - What Percent of Actual Positive values were correctly classified.

Recall = $True\ Positive/(True\ Positive + False\ Negatives)$
= 810/(810+180)
= 9/11
= 0.81

- F1-Score - Harmonic Mean of Precision and Recall and it is convenient to combine the performance of classifier into a single metric

F1-Score = $(\ 2\ X\ Precision\ X\ Recall)/(Precision + Recall\ )$
= (2* 0.81 * 0.81) / (0.81+0.81)
= 1.3122 / 1.62
= 0.81

Chapter 5

CONCLUSIONS

5.1 Conclusion:

In this project, a fire and human detection method based on motion detection is presented. We have combined the concept of two different papers i.e, detection of fire and detection of human beings which have not been experimented before. This project is also performing well with 81% efficiency when compared with projects related to similar subjects. For fire detection, we used image processing in a colour space YCbCr which extracts the luma component from the image and processes it further to select the fire pixel and predict the output.

The real-time video testing has proved its effectiveness and validity and observed that slight changes can make it run more efficiently and in a better way so that it can be used in safety and surveillance. We have used image processing for human detection and fire detection and simulated the algorithm in python3.7. However, the modules can be written in updated versions of python for a better outcome. This technology can even be cultivated in Autobots and multicopters using deep learning for better performance and more real-time use.

After all, RISKING A DEVICE IS BETTER THAN RISKING A LIFE.

5.2 Future Works:

In this project, the following things can be integrated into various fields:

- Firstly, It can be integrated with the running cameras in various organizations to fulfil our primary task i.e to say saving lives. Also, implementing the same in the Artificial Intelligence by also integrating water pumps suppose, there is a person in flames then the primary task of the pumps will be to extinguish the fire on the person.
- Secondly, there are a lot of common cases of wildfire and I believe that the lives of animals are equally important as humans we will be modifying our project accordingly to deal with the given crisis which I believe will further elevate in future.
- The project can also be used separately, that is to say, detecting humans, which I believe can be used in drones for several projects including rescuing.

REFERENCES

[1]. Dalal, N., Triggs, B., & Schmid, C. (2021, May). Human detection using oriented histograms of flow and appearance. In European conference on computer vision (pp. 428-441). Springer, Berlin, Heidelberg.

[2]. Schalkoff, R. J. (2020). Digital image processing and computer vision (Vol. 286). New York: Wiley

[3]. R.A. Khan, J. Uddin, S. Corraya, and J. Kim, "Machine vision-based indoor fire detection using static and dynamic features," International Journal of Control and Automation, vol.11, no. 6, 2018.

[4]. J. D. Burnett and M. G. Wing, "A low-cost near-infrared digital camera for fire detection and monitoring," International Journal of Remote Sensing, vol. 39, no. 3, pp. 741–753, 2018.

[5]. Viola, P., Jones, M. J., & Snow, D. (2017). Detecting pedestrians using patterns of motion and appearance. International Journal of Computer Vision, 63(2), 153-161.

[6]. Barnouti, N. H., Al-Dabbagh, S. S. M., Matti, W. E., & Naser, M. A. S. (2016). Face Detection and Recognition Using Viola-Jones with PCA-LDA and Square Euclidean Distance. International Journal of Advanced Computer Science and Applications (IJACSA), 7(5), 371-377.

[7]. Dalal, N., & Triggs, B. (2016, June). Histograms of oriented gradients for human detection.

[8]. Sebastian Montabone and Alvaro Soto, "Human detection using a mobile platform and novel features derived from a visual saliency mechanism," Image and Vision Computing, vol.28, pp.391-402, 2010.

[9]. J. Seebamrungsat, S. Praising, and P. Riyamongkol, "Fire detection in the buildings using image processing," in Proceedings of the 2014 Third ICT International Student Project Conference (ICT-ISPC), pp. 95–98, IEEE, Bangkok, Thailand, March 2014.

[10]. Szeliski, R. (2010). Computer vision: algorithms and applications. Springer Science & Business Media.

[11]. Xiaoyu Wang, "A HOG-LBP human detector with partial occlusion handling," IEEE International Conference on Computer Vision,2009

[12]. Celik, T., & Demirel, H. (2009). Fire detection in video sequences using a generic colour model. Fire Safety Journal, 44(2), 147-158.

[13].Çelik, T., Özkaramanlı, H., & Demirel, H. (2007, September). Fire and smoke detection without sensors: Image processing based approach. In 2007 15th European Signal Processing Conference (pp. 1794-1798). IEEE.

[14]. Papageorgiou, C. P., Oren, M., & Poggio, T. (1998, January). A general framework for object detection. In Sixth International Conference on Computer Vision (IEEE Cat. No. 98CH36271)(pp. 555-562). IEEE

[15]. Qiang Zhu, Shai Avidan, "Fast human detection using the cascade of histograms of oriented gradients," IEEE Computer Society Conference on Computer Vision and Pattern Recognition,2006

Database references:

1. https://www.kaggle.com/datasets/megantors/test-images-fire-humans
2. https://www.kaggle.com/datasets/phylake1337/fire-dataset