COMBUSTION AND FLOW DIAGNOSTICS
PROJECT: YCBCR BASED OPTICAL REAL-TIME FIRE DETECTOR

PREM SAGAR S - AE14B021

1. Motivation

Conventional fire detection is done using smoke alarms which in turn use humidity or temperature sensors. This requires the sensor to be in close proximity or in the vicinity of the burning region. This is not realistic if the area of concern is large and requires too many sensors. It is neither economical nor practical. A good example is a forest fire.

With advancements in cameras and microcontrollers, it is possible to employ algorithms to 'visually' detect fire. These can reduce false alarms that may have been generated by conventional humidity or temperature sensors. The idea behind this 'computer vision' technique is to exploit the characteristics of burning flames. This includes the relationship between flame colour and its temperature. Flames emit light of certain range of colour and releases heat depending on what is being burnt and what the products of combustion will be. General fire characteristics may be stated as follows:

- Usually reddish, color changes with temperature
- At low temperature, colour ranges from red to yellow
- Usually white at high temperature
- a low-temperature flame emits a light of high color's saturation and a high-temperature flame emits a low-saturation light.
- Saturation is higher when there is extra light source or during day time and lower in the dark.
- Very high temperature flames display blue colour depending on the material.
- Flame is a gaseous phenomena. It is affected by air flow around it and may tend to change shape or oscillate accordingly.

2. Objectives

- Early fire detector - Interested in detecting a growing flame and not the core. Usually the core is of high temperature and white in colour. It will also have high saturation and hence may generate a false recognition.
- The goal is to detect an initial fire in real time in a closed space with a fixed camera position. The actual implementation can be done using microcontrollers like Arduino or Raspberry-pi. For demonstration purposes, this will be done using Matlab and a laptop webcam.
- Ultimate intention is to generate a warning signal for a growing fire rather than obtain actual fire contours from an image.
- Detection of blue flames resulting from clean combustion is not of interest as they are mostly done in controlled environments.

The objective of detecting the fire pixels is achieved by using a generic rule based color model. There are multiple color models based on RGB space, YUV space and YCbCr space. YCbCr color model is of particular interest because it has been observed that chrominance C is unaffected by luminance Y to a certain extent. Thus, this tends to eliminate false recognition of fire due to high luminance which is the case with an RGB color model.

3. YCbCr based algorithm

- Y - Luminance is a photometric measure of the luminous intensity per unit area of light travelling in a given direction. It describes the amount of light that passes through, is emitted or reflected from a particular area, and falls within a given solid angle.
- C - Chrominance (chroma or C for short) conveys the color information of the picture, separately from the accompanying luma signal (or Y for short)

FIGURE 3.1. Image containing fire

FIGURE 3.2. R channel

FIGURE 3.3. G channel

FIGURE 3.4. B channel

FIGURE 3.5. Y channel on grayscale 0-255

FIGURE 3.6. Cb channel on grayscale 0-255

FIGURE 3.7. Cr channel on grayscale 0-255

- By taking advantage of the fact that chrominance information does not suffer from the illumination changes, it would be more reliable to establish the color model for the fire-pixel detection on the chrominance domain, rather than a mixture of illuminance and chrominance, such as RGB.
- It must also be noted that in the core regions of the fire Cr component is lower than the periphery of the fire. Hence, it is tough to tell apart any other region with this region. Hence, only detection of periphery of fire is pursued.
- The mean of the luminance and chrominance of an image is defined as follows. M and N - number of pixels in X and Y directions respectively.

$$
\begin{aligned}
\hat{Y} &= \frac{1}{M \times N} \sum_{x=1}^{M} \sum_{y=1}^{N} Y(x,y), \\
\hat{Cb} &= \frac{1}{M \times N} \sum_{x=1}^{M} \sum_{y=1}^{N} Cb(x,y), \\
\hat{Cr} &= \frac{1}{M \times N} \sum_{x=1}^{M} \sum_{y=1}^{N} Cr(x,y),
\end{aligned}
$$

- It has been observed that fire pixels saturate in the R channel. This can be translated to the YCbCr space as the following rules. Statistically, the luminance component is greater than the Chrominance-blue component and Chrominance-red component is greater than the Chrominance blue component. (1 - potential fire pixel, 0 - not a fire pixel).

$$
R_1(x,y) = \begin{cases} 1, & \text{if } Y(x,y) \geq Cb(x,y); \\ 0, & \text{otherwise.} \end{cases}
$$

$$
R_2(x,y) = \begin{cases} 1, & \text{if } Cr(x,y) \geq Cb(x,y); \\ 0, & \text{otherwise.} \end{cases}
$$

- Flames are usually the brightest region in the fire. These pixels will have a higher luminance than the mean luminance of the image, a lower Chrominance-blue than the mean Chrominance-blue of the image and a higher Chrominance-red than the mean Chrominance-Red of the image. These can be formualated as the following rule.

$$
R_3(x,y) = \begin{cases} 1, & \text{if } (Y(x,y) \geq \hat{Y}) \cap (Cb(x,y) \leq \hat{Cb}) \\ & \cap (Cr(x,y) \geq \hat{Cr}); \\ 0, & \text{otherwise.} \end{cases}
$$

- The Cb and Cr channels differ from each other by a large extent. Cb component is very low and Cr component is very high. This gives the new rule below. The parameter τ can be obtained statistically from several images and can also be used for calibration. Also if we take a closer look at the Cb and Cr channels, the difference is larger in the core region than other regions not containing fire. Thus, τ can be used to capture a certain spectrum of the core region although this may increase the chances of false detections.

$$
R_4(x,y) = \begin{cases} 1, & \text{if } |Cb(x,y) - Cr(x,y)| \geq \tau; \\ 0, & \text{otherwise.} \end{cases}
$$

- Celik et al. [1] manually segmented fire pixels of 1,000 images, and the histogram of a total of 16,309,070 pixels is created in the Cb-Cr chrominance plane . Figure 3.8 shows the distribution of the fire pixels in the Cb-Cr plane. The region of histograms containinng the fire pixels can be enclosed by three polynomial curves. They have obtained the polynomials by using least squares optimization. This gives the 5th rule to detect fire pixels based only on the chrominance channels.

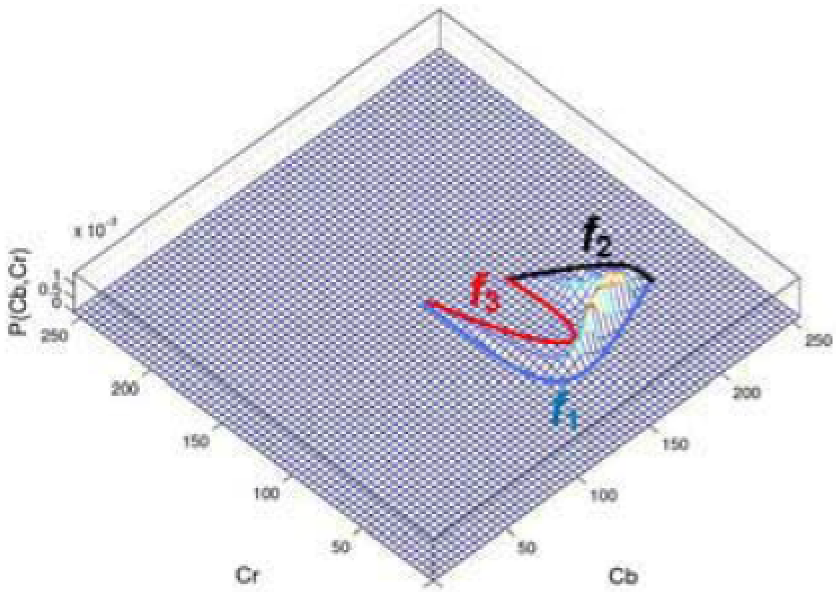


FIGURE 3.8. Fire pixels in Cb-Cr plane. [1]

$$
\begin{aligned}
f_1(Cr) &= -2.62\times10^{-10}Cr^7 + 3.27\times10^{-7}Cr^6 \\
&\quad -1.75\times10^{-4}Cr^5 + 5.16\times10^{-2}Cr^4 \\
&\quad -9.10Cr^3 - 5.60\times10^4 Cr + 1.40\times10^6 \\
f_2(Cr) &= -6.77\times10^{-8}Cr^5 + 5.50\times10^{-5}Cr^4 \\
&\quad -1.76\times10^{-2}Cr^3 + 2.78Cr^2 \\
&\quad -2.15\times10^2 Cr + 6.62\times10^3 \\
f_3(Cr) &= 1.80\times10^{-4}Cr^4 - 1.02\times10^{-1}Cr^3 \\
&\quad +21.66Cr^2 - 2.05\times10^3 Cr + 7.29\times10^4
\end{aligned}
$$

$$
R_5(x,y) = \begin{cases} 1, & \text{if } (Cb(x,y) \geq f_1(Cr(x,y))) \\ & \cap\ (Cb(x,y) \leq f_3(Cr(x,y))) \\ & \cap\ (Cb(x,y) \leq f2(Cr(x,y))); \\ 0, & \text{otherwise.} \end{cases}
$$

- The above rules have been implemented on the image 3.1. Each of them have been shown individually.

FIGURE 3.9. R1 only

FIGURE 3.10. R2 only

FIGURE 3.11. R3 only

FIGURE 3.12. R4 only

FIGURE 3.13. R5 only

FIGURE 3.14. R1 to R5

FIGURE 3.15. R1-R4

- The use R1-R4 instead of all of R1-R5 seems to be a safer option since a larger region of the fire is detected. This has been observed for multiple images (see Appendix A 5 for few). The remainder of this project will use only rules R1-R4 to stay on the safer side although it seems to detect fire like objects as well.

4. Dynamics Of Flames

- The algorithm described in the previous section can be used to mask out the fire regions in a still image.
- The motion of the fire from frame to frame in a video can be exploited to make a robust fire detector. To do the same, the frame is masked to obtain only the pixels containing the fire. The remaining regions are set to zero intensities.

- A possible metric that could measure change in these masked images from frame to frame is the **"image entropy"**. It is defined for a single channel (band 0-255) image as follows:

$$Entropy = \sum_i p_i Log_2 p_i$$

where p_i is the probability that the difference between 2 adjacent pixels is equal to i. It is a measure of contrast from one pixel to another.

- Although the definition is meant for a single channel image, the mean of all 3 channels (Y,Cb,Cr) can be calculated to obtain the entropy for setting a criteria.
- A video of fire that initially grows and then recedes was used to obtain entropy as defined above for the video frames. (This video contains only the fire on a black background. It's framerate is 25fps)

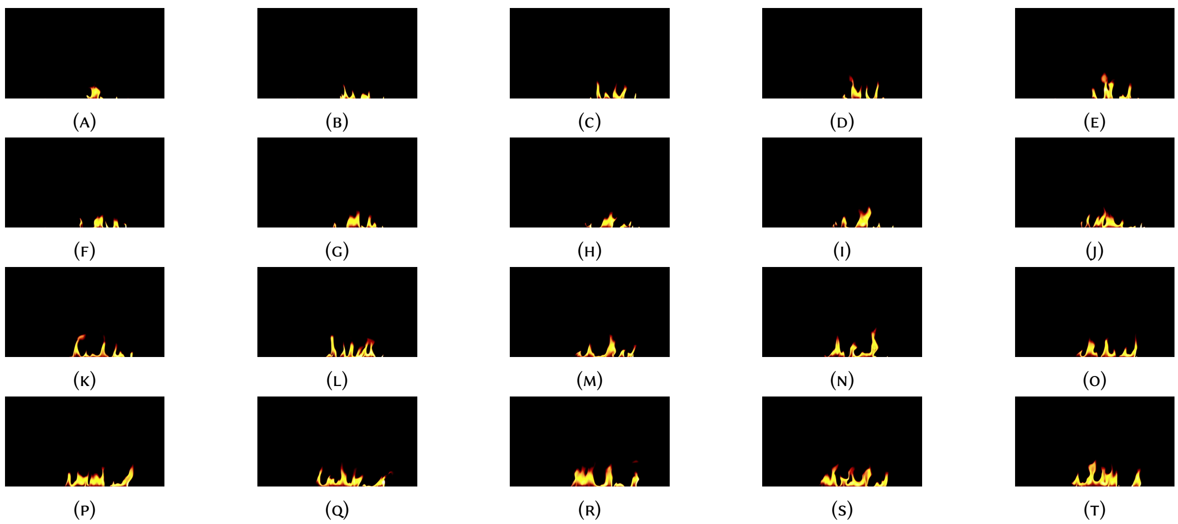


FIGURE 4.1. Snapshots of the video from frame 1 to 191 each spaced 10 frames apart

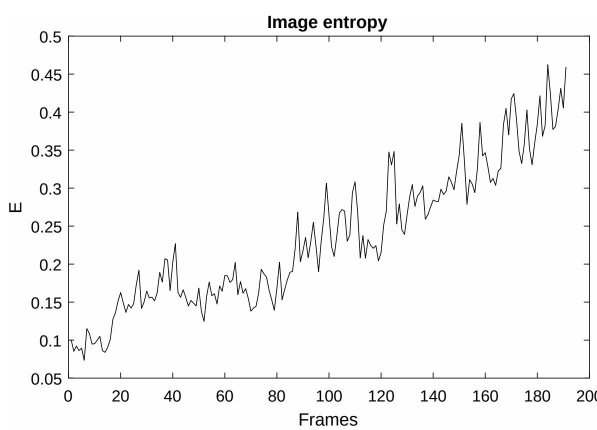


FIGURE 4.2. Entropy of masked images. Entropy increases with growing fire. Oscillations are due to fluttering of the flame

- Entropy was calculated for mulitple masked images containing fire. There is no specific trend about the entropy itself as such just as how there were trends in Y and C channels. But since a flame is dynamic, the fire pixel area changes or it's colour distribution changes or both (mostly). Thus, entropy changes from frame to frame. The following plot shows the entropy change from frame to frame.

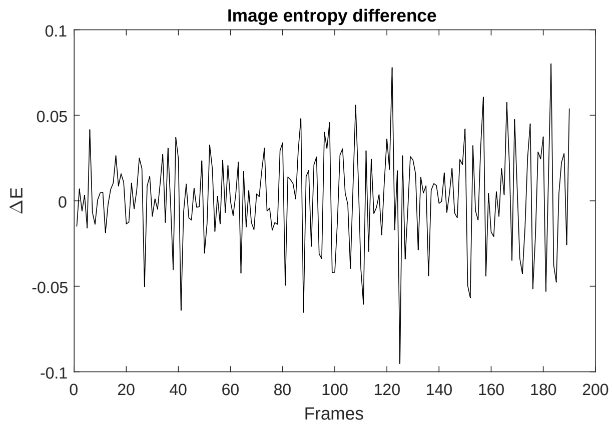


FIGURE 4.3. Frame to frame entropy difference of masked images

- The difference in entropy might be a measure of much the area and colour distribution has changed. But to set a generic rule, the difference might be too vague as it may be sensitive to conditions of the fire environment or nature of the fire. One possible solution is to measure a relative change in entropy difference and set a theshold μ to it. (A similar way of quantifying motion is used in Chen et al [2], where they use "fire disorder" defined as the difference between "flame images". This definition is vague and "fire disorder" is supposedly an array which might pose problems when setting a scalar threshold)

$$\left|\frac{\Delta E_t - \Delta E_{t-1}}{\Delta E_t}\right| \geq \mu$$

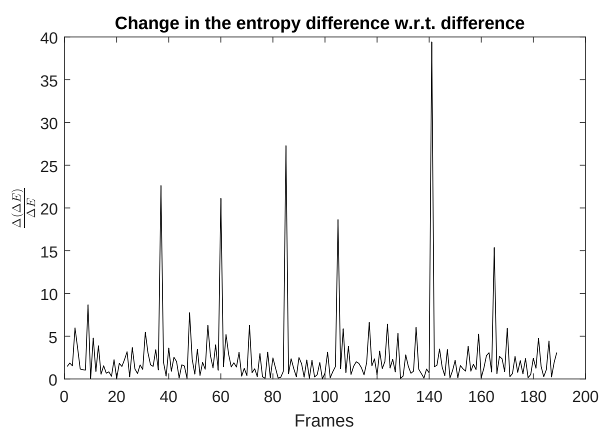


FIGURE 4.4. Relative change in entropy difference. A positive fire can be confirmed with $\mu = 3$

- The data for complete video and some other videos can be found in Appendix B 5
- The threshold μ may be dependent on a lot of things. Basically, motion of flame in 3 successive frames is in question when the relative change in entropy difference is concernced. Thus,
  - Frame rate might influence the amount of motion caught.
  - The motion itself might be dependent on the type of fire or what constituted the fire, nay, the fire burning rate which is characteristic to a specific fuel.

  - The environment of the fire may also influence the amount of airflow around the flame that causes it to fluctuate.
  - The minimum area of the field view caught on fire that is considered to be critical to generate an alarm is another parameter.
  - The area of the field of view to detect fire. For example, if a large area of a forest is the field of view, fire pixel spread rate may be small as compared to a smaller field of view like a small room.
  - Thus, μ has to be obtained by trial and error simulating a fire in the region of interest.
- A point to be noted is that fire like object usually have uniform color intensity distribution in contrast to fire flames. So if the size of the fire like object does not change much (not neglecting it's motion), it's contribution to entropy change will be lesser than an actual fire in the field of view. This should compensate for not using Rule 5.
- The algorithm was performed on multiple videos containing fire in a large forest area. (See Appendix B 5) A threshold in the range of $\mu = 3 - 6$ for a frame rate of 29.97 fps seems good.

5. Conclusions

An effective and reliable optical real time fire detector is programmed on Matlab. However, the algoritm is sligthly computationally intensive for a higher level programming language like Matlab. This is a serious concern if the response time between an actual fire event and alarm trigger is critical. The resolution of the camera is also an important factor as far as the calculation time is concerned. Currently, the response time on my computer is about 4-5 seconds for a 720p 1280x720 webcam. Actual implementation can be done on microcontrollers like Arduino or Raspberry pi where a higher level language program (preferably C++) is compiled to an assembly level language on a computer and only the assembly level language is fed to the microcontroller. This will have lower response time. It can then be integrated with an intelligent system comprising of secondary detectors like humidty or temperature sensors to confirm the fire and reduce false alarms.

References

[1] Computer Vision Based Fire Detection in Color Images - Turgay Celik and Kai-Kuang Ma.
[2] An Early Fire-Detection Method Based on Image Processing - Thou-Ho (Chao-Ho) Chen, Ping-Hsueh Wu, and Yung-Chuen Chiou.

Appendix A

(a) Image containing fire (b) R1 to R5 (c) R1-R4

Figure 5.1

(A) Image containing fire

(B) R1 to R5

(C) R1-R4

FIGURE 5.2

(A) Image containing fire

(B) R1 to R5

(C) R1-R4

FIGURE 5.3

(A) Image containing fire

(B) R1 to R5

(C) R1-R4

FIGURE 5.4

(a) Image containing fire (b) R1 to R5 (c) R1-R4

Figure 5.5

(a) Image containing fire (b) R1 to R5 (c) R1-R4

Figure 5.6

(a) Image containing fire (b) R1 to R5 (c) R1-R4

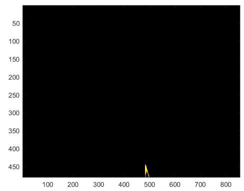
Figure 5.7

From these figures, it is seen that using rules R1-R4 is safer than using rules R1-R5

Appendix B

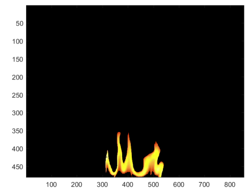
Video 1 : The entropy plots for the entire video mentioned in 4 . Since the video background is black, it is easy to evaluate what regions of the fire is being detected. Both the original frame and the algorithm masked images are shown.

(a)



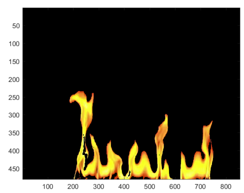
(b) Mask

(c)



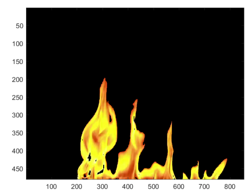
(d) Mask

(e)

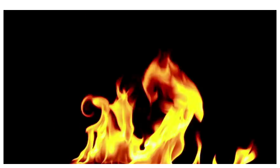


(f) Mask

(g)

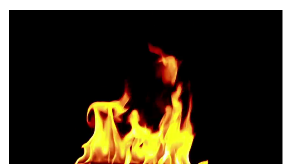


(h) Mask



(i)

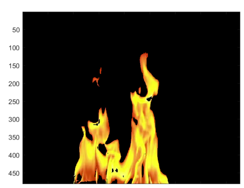
(j) Mask



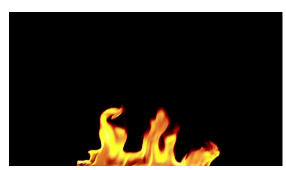
(k)

(l) Mask

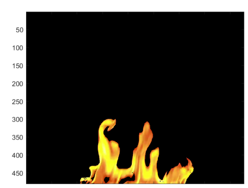
(m)



(n) Mask



(o)



(p) Mask

(q)

(r) Mask

Snapshots of the video from frame 1 to 2000 each spaced about 450 frames apart

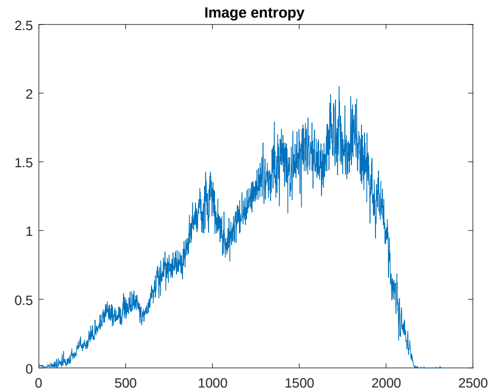


FIGURE 5.8. Entropy of masked images. Entropy increases with growing fire. Oscillations are due to fluttering of the flame. The fire initially grows and then recedes.

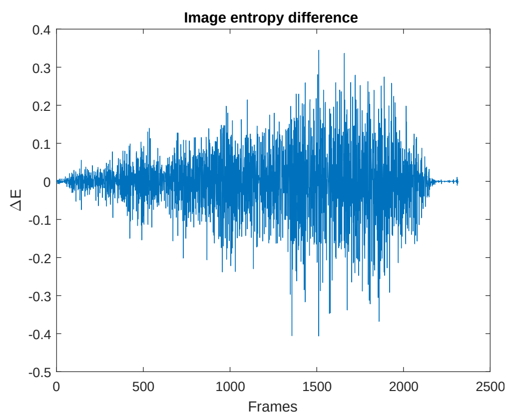


FIGURE 5.9. Frame to frame entropy difference of masked images

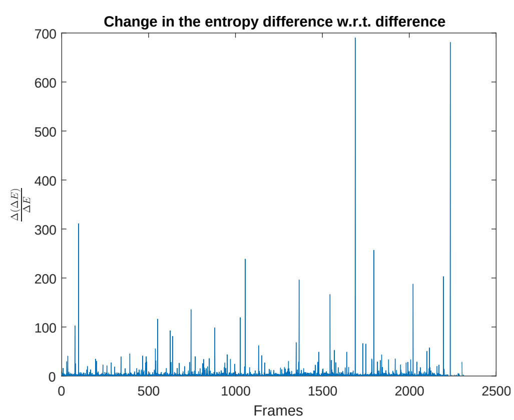
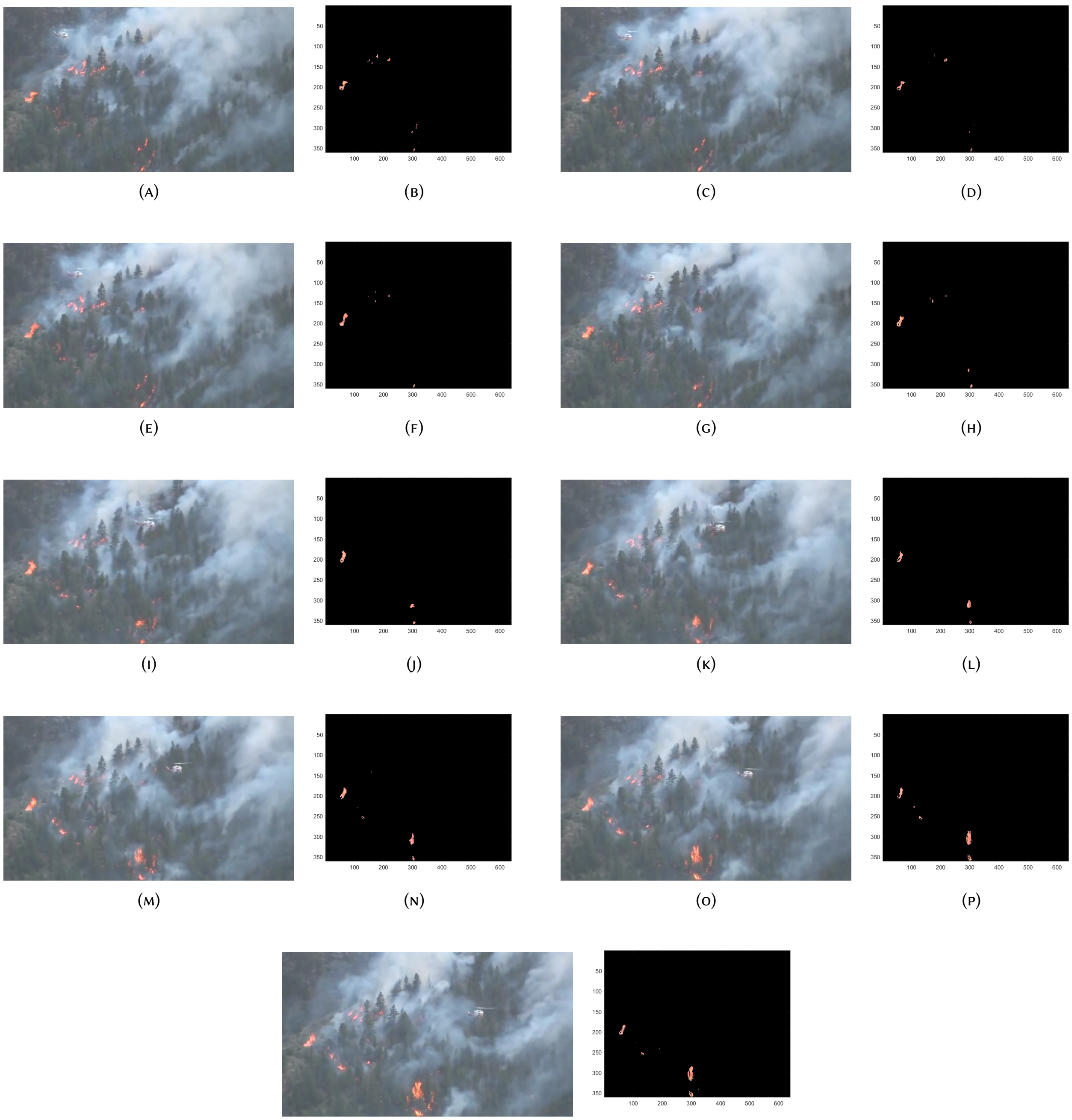


FIGURE 5.10. Relative change in entropy difference. A positive fire can be confirmed with $\mu = 3$. The large peaks clearly indicate sudden changes in entropy.

Video 2: Another video of a forest fire that is shot from a fixed camera at 29.97 fps. Slight oscillations are present in the video because of the wind shaking the camera. The alogrithm performs good to detect the fire regions. A threshold $\mu = 5.$ can be set to positively confirm fire for this set up.



(A) (B) (C) (D)

(E) (F) (G) (H)

(I) (J) (K) (L)

(M) (N) (O) (P)

(Q) (R)

Snapshots of the video from frame 1 to 450 each spaced 50 frames apart

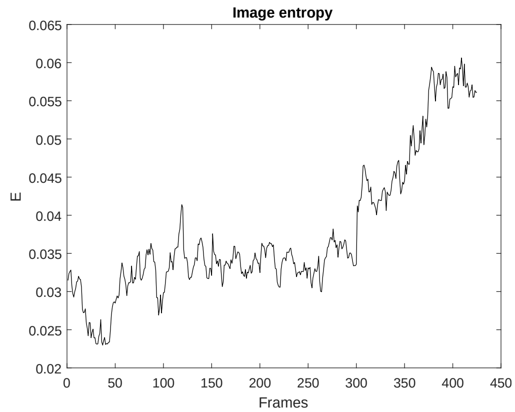


FIGURE 5.11. Entropy of masked images for Video 2. Entropy increases with growing fire. Oscillations are due to fluttering of the flame.

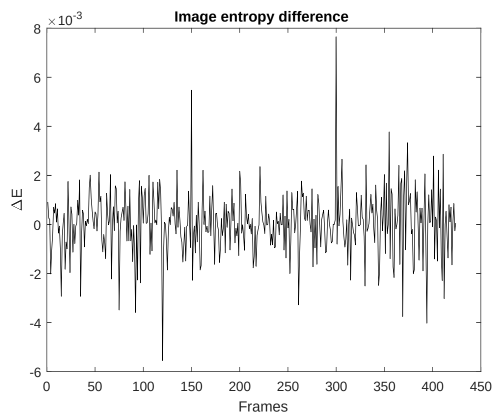


FIGURE 5.12. Frame to frame entropy difference of masked images

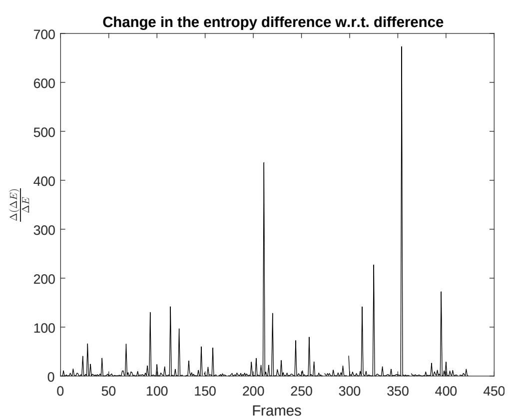
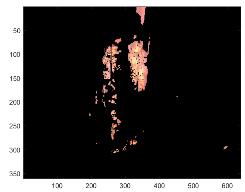


FIGURE 5.13. Relative change in entropy difference. Here again a positive fire can be confirmed with $\mu = 5$. The large peaks clearly indicate sudden changes in entropy.

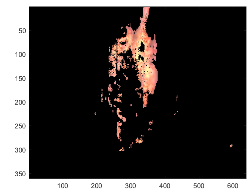
Video 3: Another video of a forest fire that is shot from a fixed camera at 29.97 fps. . The alogrithm performs good to detect the fire regions. A threshold $\mu = 5.$ can be set to positively confirm fire for this set up.

(A)



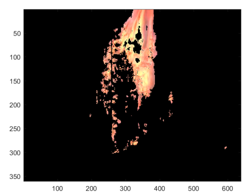
(B)

(C)



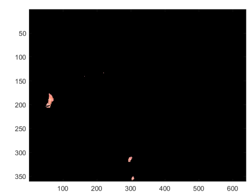
(D)

(E)



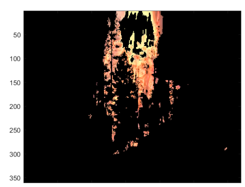
(F)

(G)



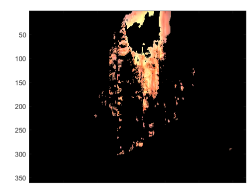
(H)

(I)



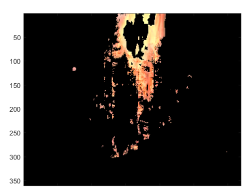
(J)

(K)



(L)

(M)



(N)

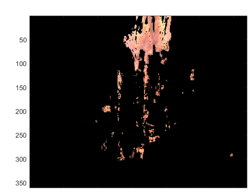
(O)

(P)

(Q)

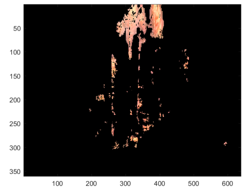
(R)

(S)



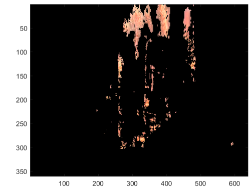
(T)

(u)



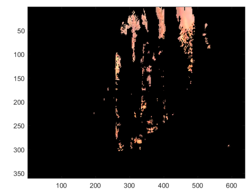
(v)

(w)



(x)

(y)



(z)

Snapshots of the video from frame 1 to 450 each spaced 50 frames apart

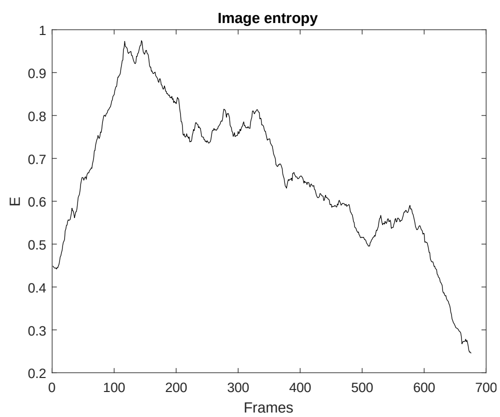


FIGURE 5.14. Entropy of masked images for Video 2. Entropy increases with growing fire. Oscillations are due to fluttering of the flame. Fire initially grows and then recedes.

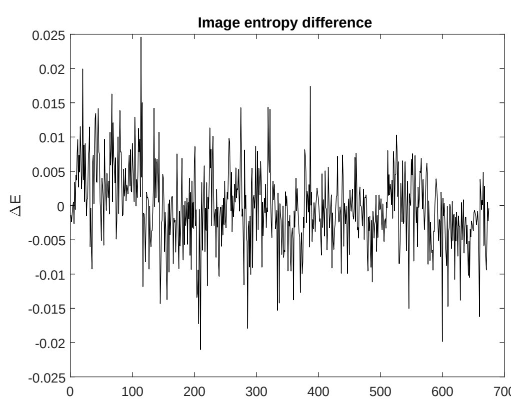


FIGURE 5.15. Frame to frame entropy difference of masked images

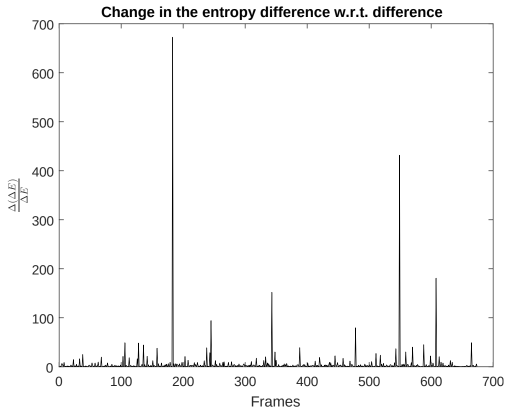


FIGURE 5.16. Relative change in entropy difference. Here again a positive fire can be confirmed with $\mu = 5$. The large peaks clearly indicate sudden changes in entropy.