

# Machine Learning Approach for Adaptive Automated Smoke Region Detection

**Master Thesis**

**Author(s):**

Ahmed, Amr

**Publication date:**

2015

**Permanent link:**

<https://doi.org/10.3929/ethz-a-010571863>

**Rights / license:**

In Copyright - Non-Commercial Use Permitted



Eidgenössische Technische Hochschule Zürich  
Swiss Federal Institute of Technology Zurich

# Machine Learning Approach for Adaptive Automated Smoke Region Detection

Master Thesis

Amr Ahmed

October 21, 2015

Advisors: Prof. Dr. Joachim Buhmann, Dmitry Laptev  
Department of Computer Science, ETH Zürich



---

## Abstract

Automatically detecting smoke and fire in natural scenes can potentially lead to forest fire prevention and therefore to saving millions of budget money. In this thesis, we develop a novel algorithm that detects potential smoke regions in images. Data comes in the form of sequences of three images (image triplets) of the each scene with a bounding box highlighting the smoke whereabouts. The algorithm extracts features from these images and uses unsupervised machine learning techniques to detect potential smoke regions. First, The algorithm divides images into superpixels, extracts features per superpixel and then clusters the superpixels in such a way that separates smoke regions from non-smoke ones.

The novelty of the algorithm comes from the fact that it operates in a semi-supervised manner due to the absence of exact labels. It is built in a modular way that is easy to debug and extend. It doesn't rely on fixed thresholds which makes it independent of specific datasets and thus can generalize to new, unseen conditions.





---

## Acknowledgement

---

I am heartily thankful to all those who supported me in any respect toward the completion of my Master thesis. I owe my deepest gratitude to my supervisor Dmitry Laptev who supported me with his guidance and rich knowledge. My gratitude also goes to Barbara Haupt who supported along the way. I would also like to thank Prof. Dr. Joachim Buhmann for giving me the opportunity to work on this topic.



---

# Contents

---

<b>Contents</b>	<b>v</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Thesis Contribution . . . . .	2
1.2 Thesis Organization . . . . .	2
<b>2 Background</b>	<b>3</b>
2.1 Supervised learning for smoke detection . . . . .	3
2.2 Manually crafted rules for Smoke detection . . . . .	5
2.3 Relation between Related Works and Our Pipeline . . . . .	6
2.4 Related Algorithms Overview . . . . .	6
2.4.1 SLIC Algorithm . . . . .	6
2.4.2 Optical Flow . . . . .	7
2.4.3 Histogram Equalization . . . . .	8
2.4.4 Expectation Maximization . . . . .	8
2.5 Dataset Description . . . . .	9
<b>3 Algorithm Description</b>	<b>13</b>
3.1 Image Preprocessing . . . . .	13
3.2 Image Segmentation . . . . .	14
3.3 Splitting Forest and Sky Regions . . . . .	15
3.4 Feature Extraction . . . . .	15
3.4.1 Intensity . . . . .	16
3.4.2 Intensity Standard Deviation (SD) . . . . .	16
3.4.3 Difference (Diff) Image Intensity . . . . .	16
3.4.4 Diff Intensity SD . . . . .	17
3.4.5 Motion Magnitude . . . . .	17
3.4.6 Motion Magnitude SD . . . . .	17
3.4.7 Correlation . . . . .	18
3.4.8 Correlation Standard Deviation . . . . .	18

## CONTENTS

---

3.4.9	Other features . . . . .	18
3.4.10	Note on Feature Choice . . . . .	19
3.5	Combining All Together . . . . .	19
3.6	Extending the Algorithm to Image Sequences of Arbitrary Length . . . . .	21
3.7	Another Variant of the Algorithm . . . . .	21
<b>4</b>	<b>Experiments and Results</b>	<b>29</b>
4.1	Dataset Split . . . . .	29
4.2	Evaluation Metrics . . . . .	29
4.2.1	Hit Rate . . . . .	29
4.2.2	False Positive Rate (FPR) . . . . .	30
4.2.3	Other Metrics . . . . .	30
4.3	Experimental Setup . . . . .	31
4.3.1	Determining best feature configuration in Forest Region	31
4.3.2	Determining best feature configuration in Sky Region	33
4.3.3	Combining forest and sky configurations . . . . .	34
4.4	Visualization Of results . . . . .	38
4.5	Discussion . . . . .	38
<b>5</b>	<b>Conclusion</b>	<b>43</b>
5.1	Model limitations and Future work . . . . .	43
	<b>Bibliography</b>	<b>47</b>

## Chapter 1

---

# Introduction

---

Forests represent an important part of Earth's ecological system. They regulate Earth's climate and serve as a habitat for millions of animals and plants' species. Unfortunately, thousands of square kilometres are damaged by forest fires every year. According to [9], in the US alone there is between 60,000 - 240,000 forest fires burning between 12,000 to 40,000 square kilometres per year.

The problem with forest fires is that forests are usually remote, abandoned areas filled with trees, dry and parching wood, leaves that act as a fuel source [11]. Fire ignition can be caused by several factors like human intervention (barbecues, camp-fires) or natural sources like lightening or hot temperatures. After ignition, forest fire spreads in an uncontrollable matter due to the abundance of fuel sources and since forest locations are usually remote, forest fires are usually detected at a late stage after damaging large areas.

In general, smoke usually appears before flames. This means that sensors that can detect smoke can be used to prevent forest fires early [5]. Conventional indoor smoke detection techniques such as infra-red sensors, particle sampling, air transparency testing can not be used to detect forest smoke. This is due to the large distance between the point of installation of these sensors and the smoke position. In order for these techniques to be effective in open areas, smoke should fill a large fraction of these areas. Another technique is to have human observers constantly observing forests through staying in towers built in the forests for that purpose. This approach is clearly costly and not scalable.

Recently, research has focused on smoke detection using surveillance cameras. Satellites have been used to monitor forests but weather presents a main limitation for this approach since image quality is severely affected by weather conditions [11]. Another approach is to use CCD (charge-coupled

device) cameras to continuously monitor forest regions. Such cameras capture video footage or static images and send them to a computer program in order to detect smoke. This is clearly a cheap and scalable approach since cameras can monitor large areas and can be installed at numerous locations. However, it still faces some challenges. First, Smoke usually doesn't have a fixed shape and thus it is hard to apply standard object recognition techniques. Second, smoke can have various colours ranging from very light gray to very dark gray in grayscale images. In addition, smoke is transparent and doesn't completely block the background from the view. Such challenges make smoke detection a challenging task. However, existing research currently focuses on smoke properties such as its motion or its tendency to blur background objects to tackle the problem.

### 1.1 Thesis Contribution

In this thesis, we aim to detect smoke regions in static images of forest fires in a semi-supervised manner. In order to do so, we develop an extensible pipeline that divides incoming images to superpixels, extracts features from these superpixels, excludes non-smoke areas based on these features until only smoke areas are left. The pipeline is easy to understand since it operates in a modular way. At each round, it chooses a new feature, and excluded non-smoke regions based on this feature until only smoke regions are left. This makes it easy to debug and add/remove new features.

The Novelty of this work comes from the fact that we are solving the task for static (non-video) images. In addition, we solve the task in a semi-supervised manner, since the only labels we have are rough labels that shows roughly the whereabouts of the smoke in the scene. In addition, the pipeline we develop is a flexible pipeline that learns from the data using machine learning techniques instead of using fixed thresholds. This makes the pipeline adaptive to unseen conditions.

### 1.2 Thesis Organization

In Chapter 2, the background is presented. In the Background, we present literature review of the related work and provide an overview of the algorithms used in this thesis, for example: SLIC algorithm, Optical Flow, etc. In Chapter 3, the smoke detection pipeline is described in detail together with the design decisions we made. In Chapter 4, We present the experiments, their results and how the model parameters were selected. In chapter 5, conclusion and suggestions for future work are presented.

# Background

---

The problem of smoke detection has been widely studied in literature. Most of the works aim at detecting smoke in video data and they use either rules based on fixed thresholds or supervised learning techniques. In this section, we describe some of the related approaches and focus on features they extract from images as well as the pipeline used to detect smoke. This helps us later on in designing our own pipeline to solve the problem in a semi-supervised manner on static images.

We could categorize the work of the literature to two main categories. The first category extracts features from smaller smoke blocks and used supervised learning techniques to train a classifier how to separate between smoke and non-smoke regions. The second category decides on some manual rules like an intensity range for example, and use these manually crafted rules to detect smoke regions.

### 2.1 Supervised learning for smoke detection

In [4], texture is used to characterize smoke in videos. First, each frame is divided into square blocks and foreground blocks are selected using an adaptive Gaussian Mixture Model for background subtraction. Afterwards, Local Binary Pattern (LBP) feature is used to capture the block appearance. For every pixel, LBP value is obtained by comparing pixel's intensity value to the values of neighbours. For pixel  $c$  with coordinates  $(X_c, Y_c)$ , LBP is defined as:

$$LBP_{P,R}(x_c, y_c) = \sum_{p=0}^{P-1} s(g_p - g_c) 2^p \quad (2.1)$$

where  $g^c$  is the value of the central pixel,  $g^p$  ( $p = 0, \dots, P-1$ ) represent the gray values of  $P$  equally spaced pixels on a circle of radius  $R$  ( $R > 0$ ) and  $s(x)$  is



the sign function. To compute LBP per block, LBP of individual pixels are accumulated in a discrete LBP histogram which is used as the feature vector of the block. Finally, block based features are used to train a Support Vector Machine (SVM) classifier which is further used to classify test example.

In [5], spatial-temporal visual features are used to detect forest smoke in video data. First, key frames are detected where a new frame is a key frame if the intensity difference compared to last key frame is above a certain threshold. A new key frame is partitioned into smaller blocks and moving blocks are detected. In order to detect motion, each block is compared to its corresponding block in the previous key frame and the block is declared moving if the intensity difference between the block in the current and previous key frames is above a certain threshold. Afterwards, five features are extracted per block, namely:

- Mean Intensity
- Intensity Skewness
- Average of wavelet energy
- Wavelet energy skewness
- Motion orientation

Authors chose these features based on the fact that smoke has higher intensity, frequency and an upward motion tendency compared to the background. In the final stage of the pipeline, random forest are trained using the previously extracted features and are used to test new forest videos.

In [6], a four-stage smoke-detection algorithm using video images is proposed. First stage involves segmenting moving parts in a video frame using background subtraction. Second, Fuzzy C Means (FCM) is used to cluster segmented moving parts according to color information. The clusters closest to the smoke color are selected for further processing. The extracted candidate regions are used to extract a number of feature. Centroid of a frame  $N$  is computed as mean  $x$  and  $y$  positions of the candidate region of the frame and then the following features are computed:

- Short Distance (SD): distance from the centroid of the previous frame, to the centroid of the present frame
- Long Distance (LD): distance from centroid of first video frame to the centroid of the present frame
- Alpha: the angle between the long distance vector and the vertical axis of the image.
- Mean Intensity of candidate region

- Area randomness  $|A_n - A_{n-1}|$ , where  $A_n$  is number of pixels in frame  $N$ .
- Variance of all previous features

Extracted features are then used to train an SVM classifier which is used afterwards to test frames in test videos.

## 2.2 Manually crafted rules for Smoke detection

Some works of the literature use manually designed rules based on fixed thresholds to solve the problem. In [15], slowly moving areas are detected using background subtraction techniques. Authors then perform connectivity analysis to merge pixels of the moving regions into larger blobs. Classification of moving blobs into smoke is based on how much of the blob is actually moving. Optical flow is used to detect motion within the blobs and if more than 20% of the blob area is moving in the direction of smoke (angle of  $[0,45^\circ]$  with image vertical axis), the blob is detected as smoke. Weber Contrast analysis is used to exclude moving blobs that are not smoke.

In [7], authors present a fast technique for motion detection use use it to detect smoke. Frames are segmented into smaller blocks and blocks are declared as moving if the difference of the intensity sum of a block and intensity sum of the same block in the previous frame is above a certain threshold. To make computations faster, integral image is used to calculate the block sum in a fast way. To estimate motion orientation, authors assume that objects within the same block move in the same direction. For a given block, they consider all the surrounding blocks with at most 3 pixels shift in the vertical and 3 pixels shift in the horizontal directions and try to match the considered block in the current frame to one of these blocks in the previous frames. The current block is matched to the block in the previous frame that produces that least difference of sum of block intensities. Finally, a frame is declared moving if the ratio of blocks moving upwards compared to all blocks is above a certain threshold.

In [16], smoke is detected in color video data. First, the current frame is segmented into blocks and moving blocks are extracted if their mean absolute intensity (MAD) difference is above a threshold where MAD is defined as:

$$MAD = \frac{1}{n^2} \sum_{i=1}^{n-1} \sum_{j=1}^{n-1} |C_{ij} - R_{i,j}| \quad (2.2)$$

where  $n$  is the number of block pixels and  $C_{ij}$  is the pixel of current block and  $R_{ij}$  os the corresponding pixel of the previous block.

Afterwards, smoke color is analysed. The authors classify smoke into two categories: dark smoke ranging from black-greyish to black and light smoke ranging from white-bluish to white. For the first category, red, green and blue components should be roughly similar. For this category, block is declared smoke is  $|\max(R,G,B) - \min(R,G,B)| < \text{threshold}$ . For the light smoke category, authors observed that blue is the highest component and that  $|R - G| < \text{threshold}$ . Finally, motion direction is estimated for each block, and blocks showing upward motion are declared as smoke.

### 2.3 Relation between Related Works and Our Pipeline

We noticed that most of the related works follow a similar pipeline:

- Video frames are divided into smaller blocks
- Features are extracted per block
- Supervised learning algorithm is trained using labelled block data and is used to classify new blocks

As we describe in 3, we follow a similar pipeline adapted to the task we are trying to solve

- Images are divided into superpixels instead of square blocks. Features are extracted per block
- Features are extracted per superpixel
- Algorithm is trained in a semi supervised manner to determine the best features to use and the tuned pipeline is used to detect smoke regions in test data

### 2.4 Related Algorithms Overview

#### 2.4.1 SLIC Algorithm

SLIC (Simple Linear Iterative Clustering) algorithm is an algorithm that segments an image into superpixels. Superpixels are basically a group of neighboring pixels that are homogeneous in color. This segmentation facilitates applying image processing techniques later on as it allows dealing with each superpixel as one unit in contrast to dealing with each individual pixel separately. SLIC algorithm clusters an image in the color and image plane space and produces superpixels that are roughly uniform in size, compact and respect image boundaries [3].

The algorithm is a version of K-means that clusters image pixels based on their x and y positions as well as the l,a,b values of the CIELAB color space.

The distance measure is defined as follows:

$$d_{lab} = \sqrt{(l_k - l_i)^2 + (a_k - a_i)^2 + (b_k - b_i)^2} \quad (2.3)$$

$$d_{xy} = \sqrt{(x_k - x_i)^2 + (y_k - y_i)^2} \quad (2.4)$$

$$D_s = d_{lab} + \frac{m}{S} d_{xy} \quad (2.5)$$

where  $m$  is the compactness factor that determines relative importance between distance and color measures. It controls how compact or loose the superpixel can be while  $S$  is square root of superpixel size (side length if superpixel was a square). The algorithm starts by initializing centres of superpixels and assigning nearby pixels to them. It then updates the superpixel centres and iterates till convergence.

### 2.4.2 Optical Flow

Optical Flow is the pattern of motion that describes motion of objects in the scene. It is typically depicted as a vector field describing motion (both magnitude and direction) in the scene. It can be computed using two sequential images of the same scene. In order to determine such a field, we assume brightness constancy. This means that pixel at location  $x, y$  and time  $t$  will move by  $\delta x, \delta y$  after time  $\delta t$ , more formally:

$$I(x, y, t) = I(x + \delta x, y + \delta y, t + \delta t) \quad (2.6)$$

after some derivations, we can arrive to the optical flow equation to solve as in [14]:

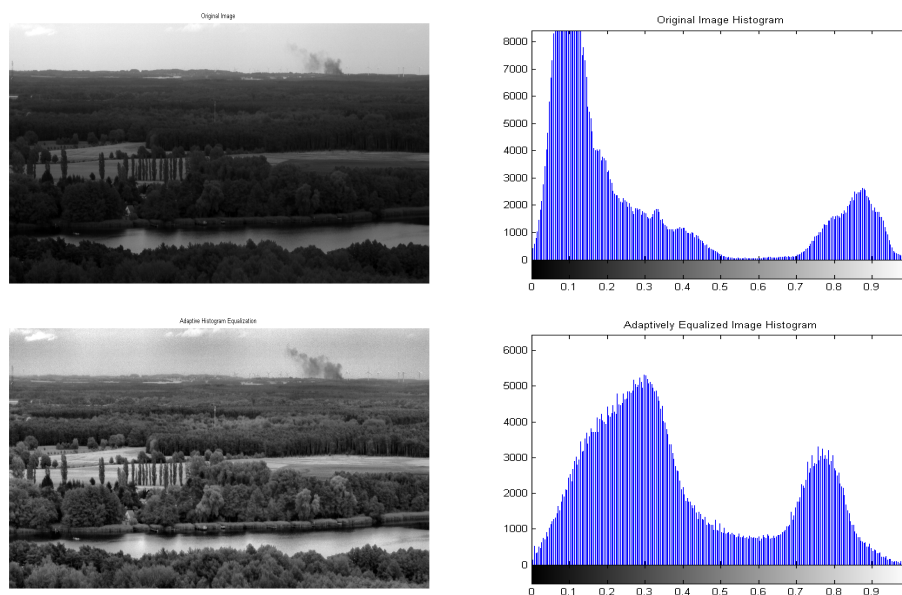
$$I_x V_x + I_y V_y = -I_t \quad (2.7)$$

where  $I_x$  is derivative of intensity w.r.t  $x$  direction,  $I_y$  derivative w.r.t  $y$  direction and  $I_t$  is derivative w.r.t. time. This is one equation with two unknowns. Several techniques exist for solving this problem such as Lucas–Kanade or Horn–Schunck methods. In this thesis, we use the optical flow implementation described in [13].

### 2.4.3 Histogram Equalization

Histogram equalization is a contrast enhancement technique that relies on adjusting the image histogram. It is used to enhance the quality of images for better visualization. In the Global Histogram Equalization technique, image intensity values are mapped to new intensity values in such a way that the resulting image histogram has a uniform distribution [1]. One problem with using the same intensity mapping for all image pixels is that it just focuses on improving the global contrast in the image and ignores local contrast of various image regions. This effect is observable in the sky for example where the local contrast between the smoke and the sky has decreased after applying the Global Histogram Equalization technique.

Another variation of Histogram Equalization is the Adaptive Histogram Equalization. In this technique, pixel intensity values are transformed according to a mapping derived from the histogram of the pixel neighbourhood. That is, the new value for the intensity of a pixel is proportional to its rank among other pixels in the neighbourhood [2]. This helps enhancing the local contrast of different image regions as in 2.1.



**Figure 2.1:** Figure showing a smoke image together with its histogram, the image after adaptive histogram equalization together with its new histogram

### 2.4.4 Expectation Maximization

Expectation maximization is an iterative technique used to estimate latent parameters in statistical models. It iterates between Expectation and Maxi-

mization steps. In the expectation step, it creates a function of the Maximum likelihood of the observed data using the parameter estimates (at first round they are randomly initialized) while in the maximization step, it uses the likelihood function to estimate model parameters that will maximize it.

In this thesis, we use Expectation Maximization for Gaussian Mixture Models. In this setting, underlying distribution of the data is assumed to be a mixture of Gaussians with each Gaussian having mean vector and covariance matrix as latent variables. The algorithm initially estimates a mean vector and covariance matrix for each cluster, computes membership weights of points belonging to clusters (probability of each point belonging to every cluster). New mean and covariance matrices are computed from the new assignment of points [12]. This is repeated until convergence, when the points assignments don't change any more.

## 2.5 Dataset Description

CCD (charge-coupled device) cameras are installed on top of towers located in various locations in German forests. Cameras capture a triplet of images for the view and then rotates to capture other angles. Typically, the camera waits 20 seconds between first and second images as well as between second and third images and then rotates 45 degrees to capture a new view. All in all, we have 189 triplets of images all of which contain smoke. 52 triplets were captured in May 2015 while the rest (137) were captured in June of the same year. The dataset features smoke situations at different times of the day as well as at varying distances from the camera locations as in the figures 2.2 and 2.3.

In addition to the triplets, ground truth for each triplet is also provided. Since exact smoke boundaries are unclear and large variance will exist anyway between human observers, ground truth was provided as rectangles surrounding the smoke area (figure 2.4). This renders the problem of detecting smoke as an instance of unsupervised or semi-supervised learning.

It is possible that the camera can shake because of wind. In this case, pixelwise comparison of images within the triplet would be useless. For this purpose, stabilization was already performed on the given dataset using a robust, subpixel accurate algorithm based on SIFT features [10]. We checked the quality of stabilization by taking every possible pair of the triplets (first and second, second and third, first and third), and for each pair, one image is shifted horizontally and vertically against the other, and the mean of the intensity difference between the image pair for each possible shift was calculated. We found that, the best shift in all pairs in all the triplets was zero which indicates that images are well stabilized.

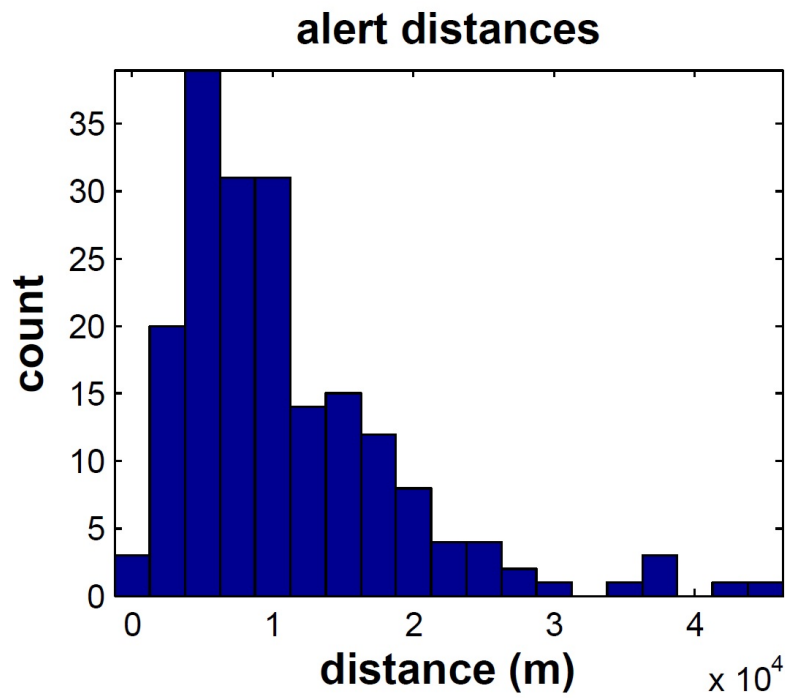


Figure 2.2: Distance between camera and smoke location

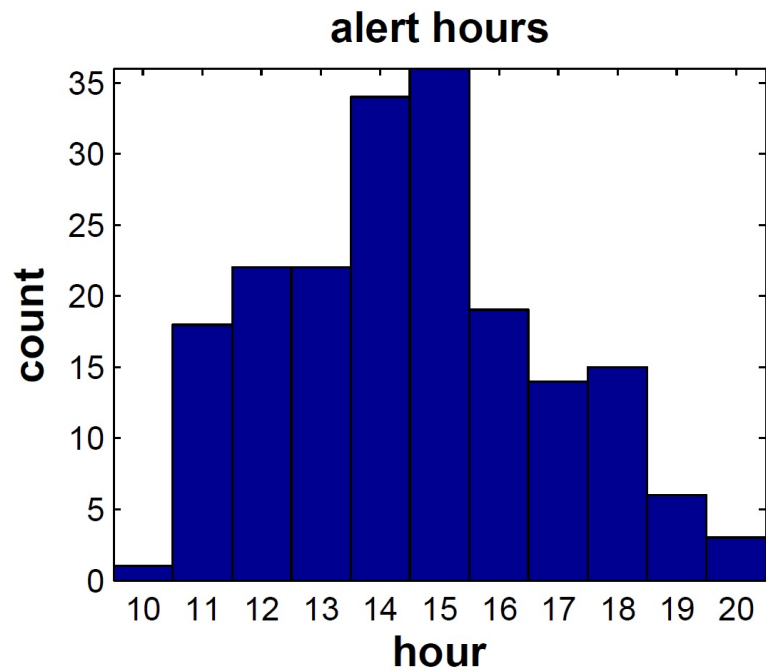


Figure 2.3: time of day at which smoke occurred



**Figure 2.4:** A triplet of images with its ground truth. First, second and third triplet images are shown at Top left, top right, bottom left respectively. Bottom right image shows the ground truth where white represents smoke region

The images consist of mainly a forest and a sky part separated by a nearly horizontal horizon line. Forests are usually dark, with the possibility of having brighter parts with no vegetation at all. As for the sky part, it is a uniform region that can have some objects like clouds, windmills or smoke.

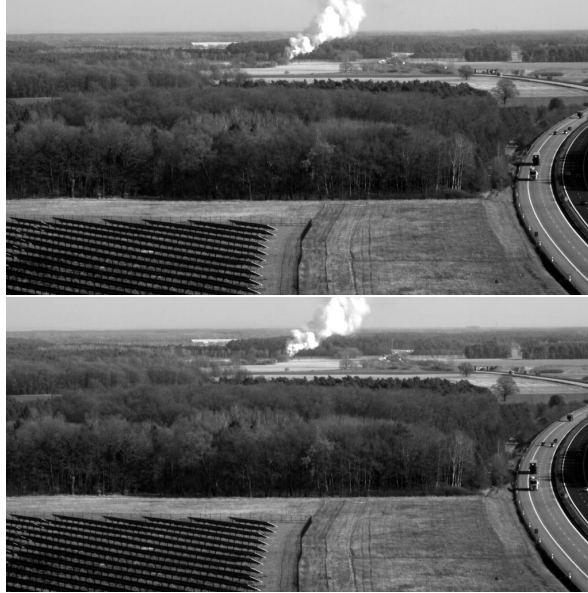
Images in the same triplet can differ in the following ways:

- Motion and expansion of smoke (figure 2.4)
- Forest motion caused by wind
- Clouds motion (very rare to occur)
- Illumination change (figure 2.6)
- Cloud shadows casted on the forest part
- Moving objects (ex: cars if there's a road or windmills in the sky) (figure 2.5)
- Movement of water (if a river, canal is present) in the forest part



## 2. BACKGROUND

---



**Figure 2.5: Moving Cars**



**Figure 2.6: Illumination Change**

# Algorithm Description

---

In this chapter, we describe how we designed the pipeline used for smoke region detection as well as the trade-offs of the design decision made. Our goal is to detect regions of smoke in static images as accurate as possible. We tuned the algorithm to make use of the fact that we have triplets of images, but it can be tuned to work on arbitrary sequences of length  $n$  where  $n > 0$ . In the following sections we describe individual parts of the algorithm and present an overview of the complete pipeline in the end.

### 3.1 Image Preprocessing

Images come in tif format with each pixel having an integer value in the interval  $[0, 65535]$ . However, a typical histogram of intensity values of an image doesn't occupy the full range. Thus we first normalize intensity values per triplet to lie in the interval  $[0,1]$  as follows:

- Merge all intensity values of all pixels in the triplet together
- Get the value corresponding to the 99.9th percentile of all intensity values from the previous step
- Divide all pixel values by the maximum value
- Set any pixel with value greater than one to one

We merge all intensity values of the triplet together and choose the maximum value from the mixed values to ensure dividing all images in the triplet by the same value. The reason of choosing the 99.9th percentile value instead of maximum value is to avoid outlier pixel intensities. This helps stretching the resulting histogram to fill the whole  $[0,1]$  range as in figure 3.1.

## 3.2 Image Segmentation

The first step in the algorithm is to segment incoming images to superpixels using SLIC algorithm. The benefits of using SLIC (compared to segmenting images into square blocks) is that superpixels are homogenous and respect image boundaries. This can be observed at the horizon line where superpixels never include simultaneously parts above and below the horizon line, an observation which allows clean separation of forest and sky regions later on.

To use SLIC algorithm, we should decide on the number of superpixels as well as the compactness factor 2.4.1. After experimenting with various segmentations, we chose number of superpixels to be 500 per image and the compactness factor to be 40. These results produced visually appealing results.

It should be noted that SLIC initially produced segmentations with rough boundaries. To solve the problem, we tried performing median filtering, gaussian filtering, median + gaussian filtering before image segmentation. We observed that filtering improved the segmentation quality by reducing roughness of superpixel boundaries 3.2. However, gaussian filtering (whether performed alone or combined with median filtering) caused blurring at the edges which results in some superpixel that only include the blurred parts of the edges. Thus we chose performing only median filtering before image segmentation.

We also experimented with the window size of the median filter and observed that it doesn't cause significant changes in the quality of the segmentation. We used a 10\*10 window size in our experiments.

One challenge with the segmentation is that if we apply SLIC individually to each of the triplet, superpixels would have different boundaries across images within the same triplet making it hard to establish correspondence between superpixels across different images in the triplet. Images of the same triplet are taken successively with a temporal difference of 20 seconds between successive images. This means that image boundaries do not change much between an image and its successor within the same triplet. We used this fact to perform segmentation only on the first image within the triplet and apply segmentation results to the second and third triplet images. This enabled establishing 1:1 correspondence between superpixels of the three images within the triplet. As we later describe, we segment each image within the triplet to superpixels, extract features from superpixels and decide based on these features whether the superpixel is smoke or not. By using same segmentation across the triplet images, we facilitate combining information obtained from corresponding superpixels in different images of the same triplet 3.3.

### 3.3 Splitting Forest and Sky Regions

In this thesis, we target detecting smoke on forest settings. Thus, it is safe to assume that incoming images would contain a bottom large part of forest with vegetation and various structures (cars, electricity towers, animals, water, etc.) and an upper relatively smaller sky region with the horizon line separating both parts. We verified this observation by looking into images of the dataset and we found that sky and forest parts are always separated by a nearly horizontal horizon line. We used this observation to separate forest from sky parts. This separation allows solving two smaller sub-problems (detecting smoke in forest and sky parts separately) instead of solving the original problem (detecting smoke in the whole image).

To perform this separation, we used the fact that sky is much brighter compared to forest and that at the horizon line, there is a sudden large drop in intensity. To detect where the drop happens, we calculated the mean intensity of pixel of each row in the image from top to bottom and plotted it on a curve. We determine the horizon to be at the rows that show the largest negative slope in the plot. We verified this method by applying it to the whole dataset and found that it produced reasonable separation of forest and sky regions as in figure 3.4.

One challenge that we faced was that the horizon line isn't exactly horizontal. While the cut produced by this method is strictly a horizontal cut. Here, we made use of the fact that superpixels respect image boundaries and that they do not cross over the horizon line and classified each superpixel in a given image segmentation to belong to the sky if most of its area lies above the boundary line or to the forest otherwise. This helped producing a cleaner separation between forest and sky regions with minor erroneous superpixels belonging to the wrong side.

### 3.4 Feature Extraction

This is the core part of the algorithm since detection of smoke relies mainly on the features extracted. Good features that provide distinction between smoke and non-smoke regions mean higher accuracy of smoke region detection. Clearly, the perfect feature would be a feature that has the value of one at smoke superpixels and zero otherwise. This would allow for the utmost separation between smoke and non-smoke regions. However, such a perfect single feature is hard to arrive at since smoke can take many forms, intensity values and doesn't have a specific shape. In this work, we investigate the use of a set of features that describe smoke properties well and can together help in smoke region detection. To assess whether the feature would be useful for our task or not, we usually inspect feature values at the region of smoke and determine whether there is some special pattern at the smoke

regions. Usually, it is the case that features we extract have relatively higher values at smoke regions compared to other regions in the image. We next describe features used, how we extracted them and why we believe they are useful for our task.

#### 3.4.1 Intensity

In grayscale images, smoke can have various intensity values ranging from very dark gray to very light shades of gray. In forest regions, we noticed that smoke is always lighter in color compared to the rest of the forest (figure 3.5). This can be attributed to the fact that forest is dark in color by default and because the type of fuel burnt (vegetation in that case). We verified this observation by looking at all images in the dataset. Thus, for forest parts we can expect smoke to be found in the lighter areas and we can safely discard darker regions. However, in the sky part, there was no clear rule for the color of smoke. In some images, smoke was found to be relatively very bright compared to the background while in others it was very dark as in figure 3.6. Thus, we use the intensity feature only in the forest part.

#### 3.4.2 Intensity Standard Deviation (SD)

This is the standard deviation of intensity values per superpixel. We expect smoke to have a relatively high intensity standard deviations due to its non-uniform nature. Within the smoke plume, we can observe that smoke exhibit different shades of grey and non-uniform texture. However, by observing heat maps of intensity standard deviation values at smoke regions compared to other regions, we found in many cases that the assumption of smoke regions having higher intensity standard deviation values was violated. Thus, we decided to exclude intensity standard deviation from features considered.

#### 3.4.3 Difference (Diff) Image Intensity

Diff image intensity is a very important feature. To calculate it, we first calculate the pixel wise absolute difference between an image pair. The value of the feature for each superpixel is the mean intensity value of the diff image at the superpixel location. Diff intensity is useful because it helps location parts that change in appearance across different images of the triplet. By observing heat maps of various diff images 3.7, we found that smoke regions tend to have relatively high diff intensity values compared to other regions. Diff intensity can be high as well for moving parts in the forest, or when the illumination changes.

It should also be noted that diff intensity can be greatly affected by the presence of noise. For this reason, we perform median filtering on images

before computing the diff image.

#### 3.4.4 Diff Intensity SD

The problem with diff intensity is that it can't differentiate between motion and intensity changes like shadows or scene illumination changes. However, the diff intensity standard deviation can tell the difference since in case of illumination changes or shadows, the pixel wise changes in the diff intensity values is uniform and thus standard deviation in these cases would be relatively of low magnitude. In figure 3.7, we can see diff intensity standard deviation values at smoke regions and it can be noticed that they are relatively high compared to other regions.

#### 3.4.5 Motion Magnitude

We used Optical flow algorithm to detect moving parts of the image. Optical flow requires as input two consecutive images and calculates motion vector the horizontal ( $V_x$ ) and vertical ( $V_y$ ) motion components. We calculate motion magnitude as follows

$$\text{MotionMagnitude} = \sqrt{V_x^2 + V_y^2} \quad (3.1)$$

Motion magnitude for a superpixel is calculated as the mean of motion magnitudes of individual pixels comprising that superpixel. We made a design decision to use motion magnitude instead of individual motion components  $V_x$  or  $V_y$  since for  $V_x$  (horizontal motion), smoke can be moving to the left or right while for  $V_y$  (vertical component), it can be the case that smoke doesn't move upwards between two consecutive images. Also, smoke moves in all directions,  $x$  or  $V_y$  components can cancel out. This is not the case with the magnitude since it is always a positive quantity. As we can see in the figure 3.8, motion magnitude tends to have higher values at smoke regions due to the moving nature of the smoke.

#### 3.4.6 Motion Magnitude SD

Despite the tendency of moving upwards, it can be observed that smoke particles move in all directions. The smoke plume has no rigid shape and it keeps changing over the course of burning. Motion magnitude standard deviation can be used to differentiate between motion of the smoke and other moving objects having more rigid shapes (figure 3.8). For example, wind blowing will cause leaves of the forest to move in the same direction. Also, rigid objects would have some low motion standard deviation since all particles of the object move in the same direction with roughly the same velocity vector.

### 3.4.7 Correlation

Correlation is a feature that measures the degree of deformation or change between two images (figure 3.9). Local correlation at pixel  $C_{ij}$  is calculated as follows:

$$C(\mathbf{x}) = \frac{\sum_{\mathbf{x}_i \in \Omega(\mathbf{x})} (I_t(\mathbf{x}_i) - \bar{I}_t(\mathbf{x})) (I_{t+dt}(\mathbf{x}_i) - \bar{I}_{t+dt}(\mathbf{x}))}{\sqrt{\sum_{\mathbf{x}_i \in \Omega(\mathbf{x})} (I_t(\mathbf{x}_i) - \bar{I}_t(\mathbf{x}))^2 \sum_{\mathbf{x}_i \in \Omega(\mathbf{x})} (I_{t+dt}(\mathbf{x}_i) - \bar{I}_{t+dt}(\mathbf{x}))^2}} \quad (3.2)$$

when  $A$  is the first image in the image pair,  $B$  is the second image,  $N$  is the window size side length of the neighbourhood used to compute the correlation value. Expectations are taken over the neighbourhood of the pixel for which correlation is computed ( $C_{ij}$ ). By substituting in the equation above with very different values (different images), we obtain a very low or negative correlation. For this reason, we redefine our correlation value as:

$$\text{New Correlation} = \min(1, 1 - \text{correlation}) \quad (3.3)$$

where correlation is the value previously computed. This yields values in the interval  $[0,1]$  with 0 meaning no change in the image and 1 meaning severe change in the image. Correlation is useful for smoke detection because smoke changes the image appearance over time. First, it moves and occupies new regions. Second, smoke blurs the background behind it. Thus, a high correlation value can be an indication of smoke. We verified this assumption by observing heat maps of correlation values.

### 3.4.8 Correlation Standard Deviation

We have visualized the standard deviation of the correlation by means of heat maps and observed that it can't differentiate well between smoke and non smoke regions. Thus, we excluded it from the set of features we consider.

### 3.4.9 Other features

In this part, we describe some of the features that we tried but didn't work. We mainly compute to check whether the feature can separate smoke regions or not. We tried to extract edges and compute intensity of edge image for each superpixel. The motivation was that smoke blurs the background and thus smoke regions would have lesser edges. However this was not completely true. Intensity varied greatly within the smoke plume. These variations caused edge operators to detect edges at smoke regions and thus

we could not differentiate between smoke and non smoke regions. Another approach we tried is to blur the image using Gaussian filter and calculate the difference in intensity before and after blurring. The motivation was that smoke has already blurred the background and thus blurring would not have a great effect on smoke regions but would severely distort non smoke regions but again by visualizing the heat map, the feature was not distinctive enough.

### 3.4.10 Note on Feature Choice

We were careful to choose features that would differentiate between smoke and non-smoke regions. Features chosen would generally have relatively higher values at smoke regions and lower values at other regions. For Forest regions, the relevant features are:

- Intensity
- Diff Intensity
- Diff intensity standard deviation
- Motion magnitude
- Motion magnitude standard deviation
- Correlation

while for sky parts, relevant features are

- Diff Intensity
- Diff intensity standard deviation
- Motion magnitude
- Motion magnitude standard deviation
- Correlation

## 3.5 Combining All Together

Having described individual parts of the algorithm we describe now the full pipeline. For a new image triplet, we segment the first image into superpixels (500 Superpixels) and use the same segmentation for second and third images in the triplet.

We divide the triplet to 3 pairs. First pair comprises first and second images. Second pair comprises second and third images while the third pair comprises first and third images. Image pairs are needed to extract features like diff intensity or motion magnitude. For every pair, we have an initial



### 3. ALGORITHM DESCRIPTION

---

image (image taken earlier in time) and latter image (taken at a later point in time). Features requiring only one image to be computed (ex: intensity) are extracted from the initial image while features requiring a pair of images (ex: diff intensity) are obtained by considering corresponding superpixels in the initial and the latter image. Thus, for a new triplet, we extract around 500 data points (SLIC implementation used computes a close number to the user specified superpixel count) from the pair of first and second images in the triplet. In this image pair, intensity is captured from the first image, while the rest of the features are captured using both images. We repeat the same procedure for the second image pair (second and third triplet images) and for the third image pair (first and third triplet pairs). This results in around 1500 data points per triplet with each data point having 6 dimensions (intensity, diff intensity, diff intensity standard deviation, motion magnitude, motion magnitude standard deviation, correlation). It should be noted that for every superpixel, there are 3 data points from the 3 pairs corresponding to it.

Now we arrive at the stage where we need to determine which of these data points correspond to smoke and which doesn't. We first split the extracted data points into two sets, one set for forest data points and the other from sky data points. We then perform the following procedure on each of the sets independently:

- Repeat for the number of rounds of current set of data points (forest or sky)
  - Pick a new feature from the feature set to be applied on the current set of data points (whether forest or sky)
  - Cluster the data points in the set into 2 clusters using the EM algorithm
  - Discard data points belonging to the cluster with lower mean value.
- Declare left data points as smoke regions

The idea behind the procedure is that we already chose features that have higher values for smoke regions and lower values otherwise. Thus by clustering into two clusters, smoke regions would be part of the cluster with higher mean value and thus at each clustering round the goal is to discard non-smoke regions so that in the end we would have the potential smoke regions left. At each clustering round, we cluster only based on one feature. This helps keeping the pipeline simple and understandable. In addition, we limit the number of clusters at each round to two clusters and only choose one cluster out of the two to limit the search space of potential configura-

tions (Other possible configurations include choosing top one out of three clusters or even top two out of three clusters based on cluster mean value).

Each superpixel in the segmentation was represented initially using three data points (one data point from each pair of images). If the superpixel is having at least one data point in the set of left data points, then this superpixel is declared as a smoke superpixel. We arrived at this rule since it is less dangerous to have false positive regions declared as smoke compared to missing the actual smoke regions.

### **3.6 Extending the Algorithm to Image Sequences of Arbitrary Length**

So far we have discussed applying the pipeline image triplets (sequences of length three). However the algorithm can be applied on a sequence of one or more images. In case of one image, some features that require a pair of images (diff Intensity, motion magnitude) won't be possible to use, however, features like intensity or intensity standard deviation can still be used. we have divided the triplet of images into 3 pairs (first and second images, second and third images, first and third images), extracted features from each pair and then combined the results in the end. This can be done also on longer image sequences, by dividing sequences into image pairs (perhaps not all possible pairs since this can be an overkill in case of longer sequences), extract features from individual image pairs, combine the results from individual pairs to decide on the regions of interest. When using a triplet of images, we decided that a superpixel is a smoke superpixel if it appeared in at least one of the image pairs as smoke. However, we can change this rule since it would mean lots of false positive regions in case of longer sequences.

### **3.7 Another Variant of the Algorithm**

The algorithm described above handles one triplet at a time. In this variant, We handle all superpixels from all triplets at once. We combine all data points of the forests of all triplets and apply clustering rounds to them and repeat the same procedure for the sky part in the same fashion as previously described (using one feature at a time, clustering into two clusters using EM algorithm and keeping the cluster with higher mean). The aim of this approach is to investigate whether combining superpixels from all triplets together can improve the clustering results.

### 3. ALGORITHM DESCRIPTION

---

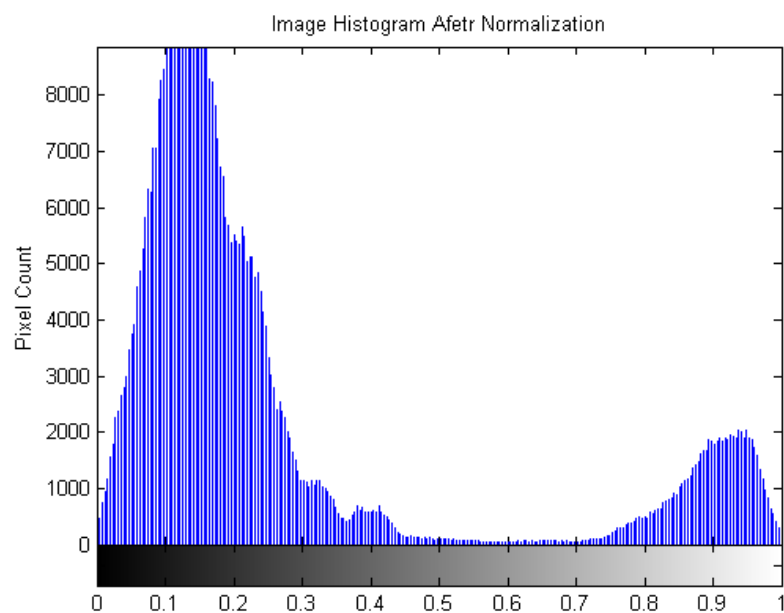
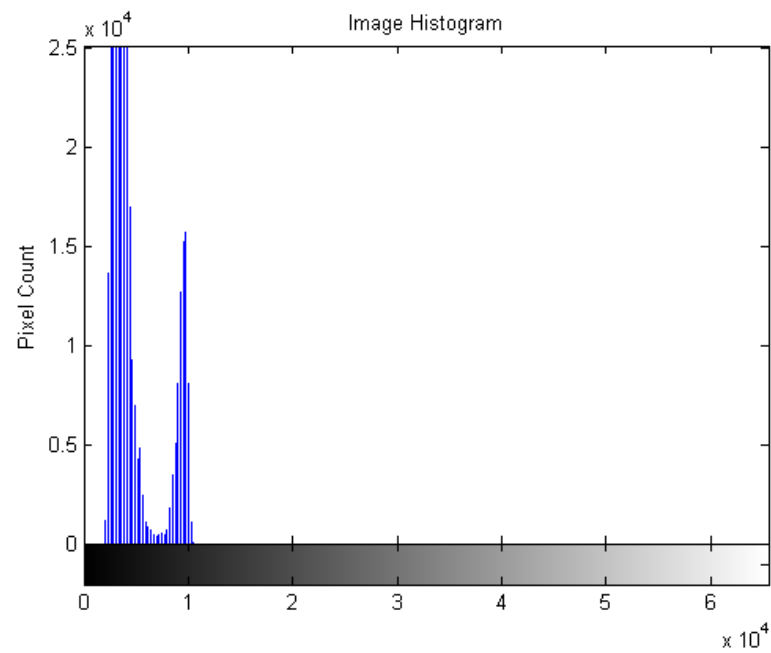
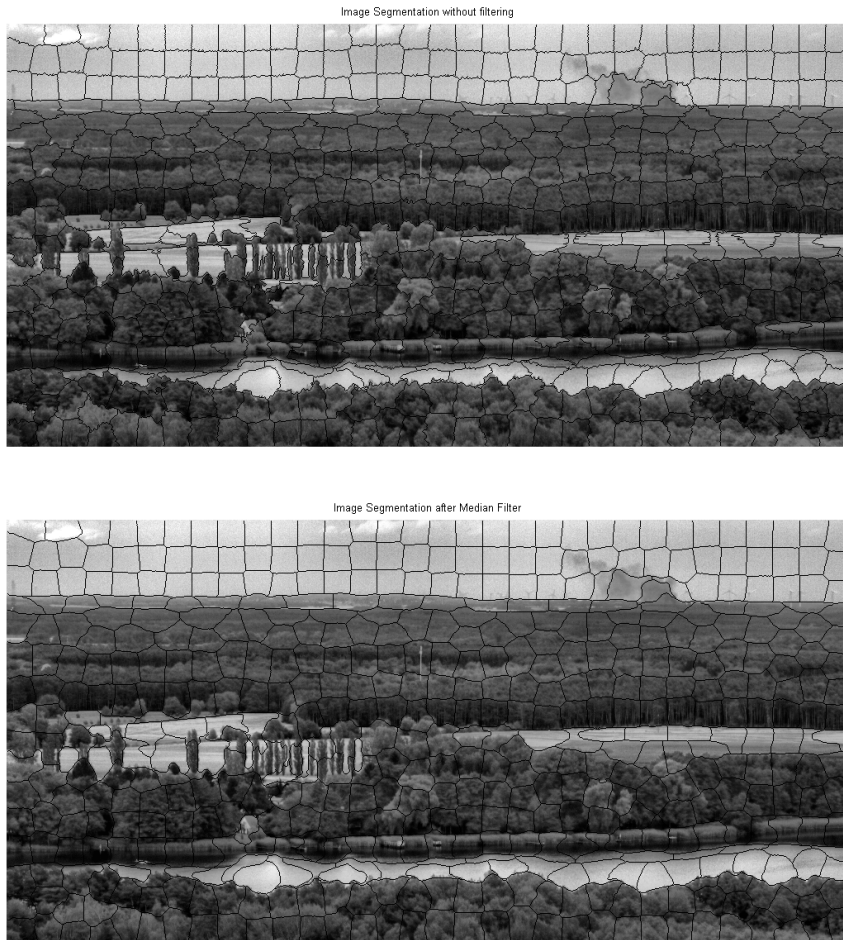


Figure 3.1: Figure showing an image with its histogram before and after normalization

### 3.7. Another Variant of the Algorithm

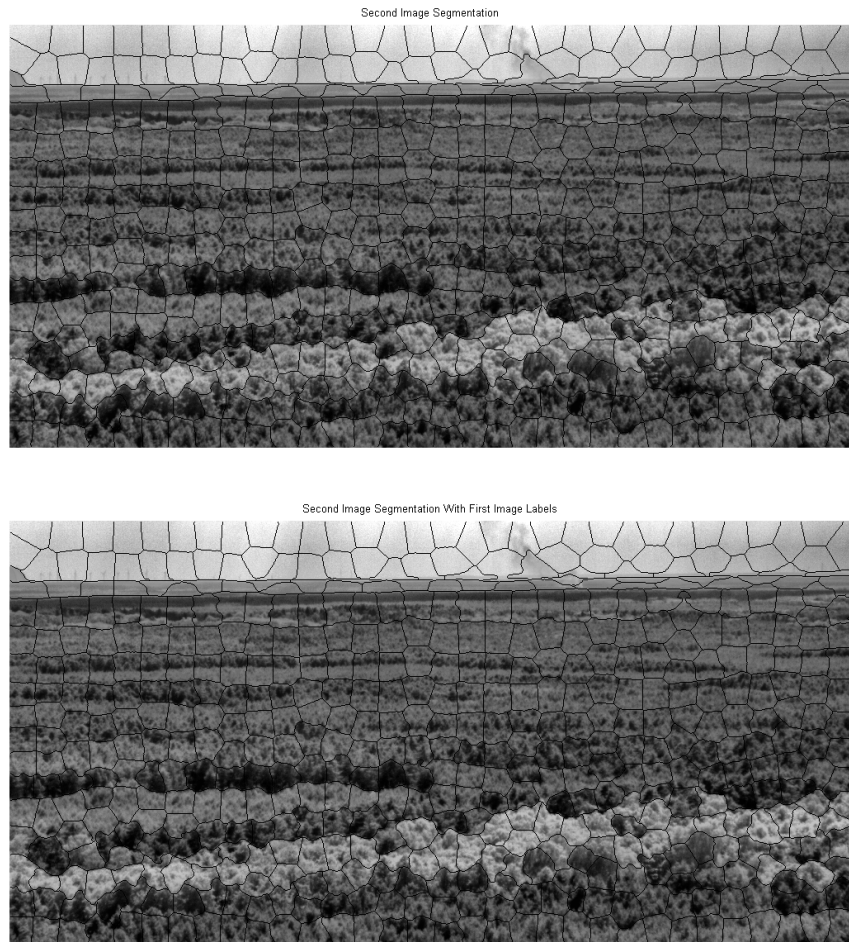
---



**Figure 3.2:** Figure showing an image segmented using SLIC algorithm without median filtering (top) and with median filtering (bottom)

### 3. ALGORITHM DESCRIPTION

---



**Figure 3.3:** Figure showing the second image of a triplet with its SLIC segmentation (left), and the same image with SLIC boundaries from segmentation of first triplet image is applied to it

3.7. Another Variant of the Algorithm

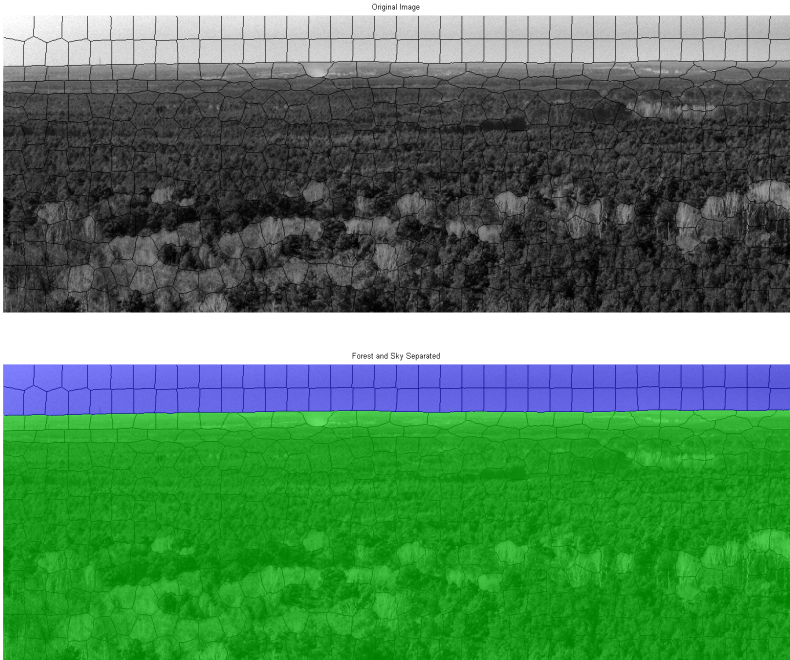


Figure 3.4: Figure showing the separation of an image to forest (blue) and sky (green) regions

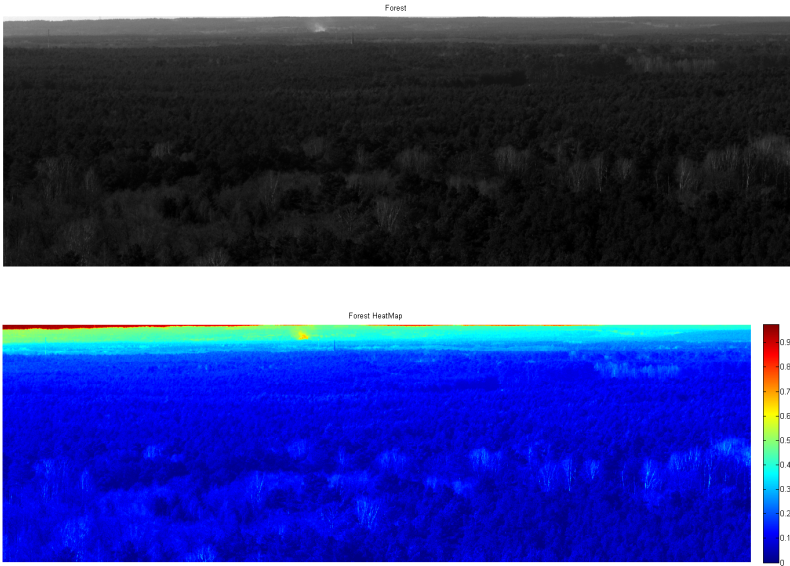


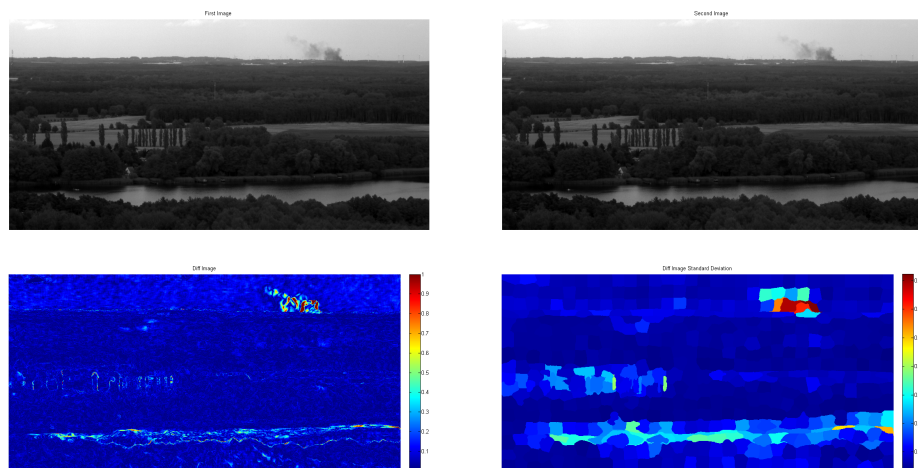
Figure 3.5: Figure showing a forest region with Smoke and its corresponding Heat map

### 3. ALGORITHM DESCRIPTION

---



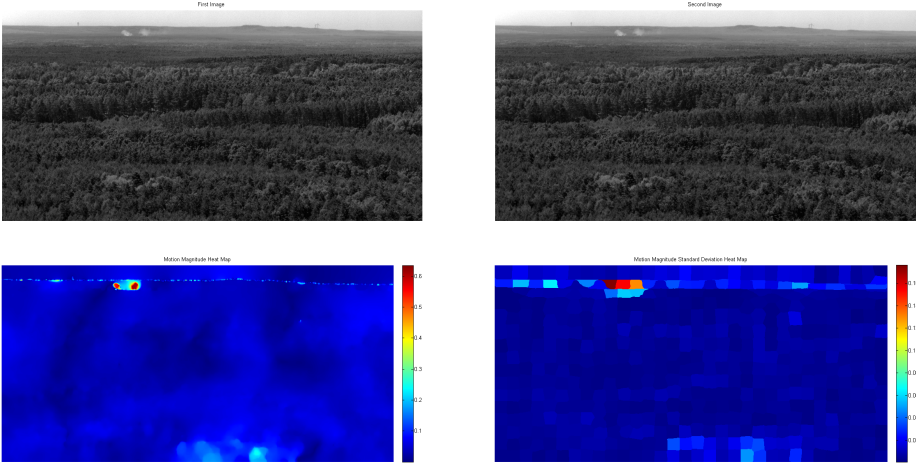
**Figure 3.6:** Figure showing instances of very bright and very dark smoke occurring in the sky



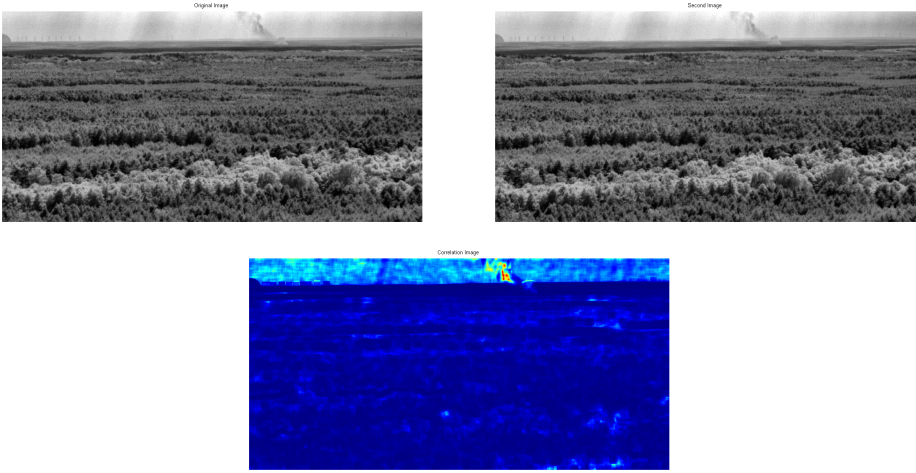
**Figure 3.7:** Figure showing two images of a triplet, first image top left and second top right, together with heat map for diff intensity (bottom left) and diff intensity standard deviation (bottom right)



3.7. Another Variant of the Algorithm



**Figure 3.8:** Figure showing two images of a triplet, first image top left and second top right, together with heat map for motion magnitude (bottom left) and motion magnitude standard deviation (bottom right)



**Figure 3.9:** Figure showing two images of a triplet, first image (top left) and second (top right), together with heat map for the correlation between them





## Experiments and Results

---

### 4.1 Dataset Split

The task at hand is a semi-supervised learning task since we have only crude labels for smoke regions. We split the data into training and test parts. For the training part, we choose the 52 triplets captured in May 2015, while the test set would comprise the 137 triplets occurring in June 2015. We tune the algorithm by choosing features to be used, number of clustering rounds on the training set and then report the results on the test set. The chosen dataset split enables us to determine whether the algorithm can generalize well to triplets captured at a later point in the future. It also allows determining the algorithm's ability to generalize to new scenes as the test set contains new scenes not included in the training set.

### 4.2 Evaluation Metrics

We considered various metrics to evaluate the accuracy of smoke region detection. Our main priority is to hit the smoke in as many images as possible. In the meantime, we aim at minimizing the area of false positive regions (non-smoke areas detected as smoke). For this purpose we defined two metrics, namely the hit rate and the false positive rate.

#### 4.2.1 Hit Rate

Hit rate is a binary variable that assumes the value of zero (smoke region isn't hit) or one (smoke region is hit) for each image triplet. This variable has value one if the intersection between the ground truth label and the regions

of interest has an area of at least an average superpixel size. More formally:

$$\text{Hit Rate} = \begin{cases} 1, & \text{if } (ROI \cap \text{Ground Truth}) \geq \text{Average Superpixel Size} \\ 0, & \text{otherwise} \end{cases} \quad (4.1)$$

where ROI stands for Regions Of Interest detected by our algorithm as potential smoke regions and

$$\text{Average Superpixel Size} = \frac{\text{Number of Pixels in Image}}{\text{Number of Superpixels in Image}} \quad (4.2)$$

#### 4.2.2 False Positive Rate (FPR)

This metric measures the false positive regions that are falsely detected as smoke. It assumes real values in interval  $[0,1]$ , where zero means that no area outside the ground truth (white box in figure 2.4) is detected as smoke while one means that all the area out of the ground truth label is falsely detected as potential smoke region. The metric is formulated as follows:

$$\text{FPR} = \frac{\text{ROI} \cap \text{Non Ground Truth Area}}{\text{Non Ground Truth Area}} \quad (4.3)$$

where the non ground truth area is the black area in the ground truth (2.4). It is clear that for achieving high accuracy of smoke region detection, we should maximize the hit rate while minimizing the FPR. The priority is given to maximizing the Hit Rate because we can not afford missing the smoke region. However, having a relatively higher FPR can be acceptable because images can then be passed to a human inspector for verification of the true smoke regions.

#### 4.2.3 Other Metrics

We considered some other metrics such as the True Positive Rate (TPR) and the Jaccard similarity. We defined the TPR as the area inside the ground truth that we could detect as smoke, more formally:

$$\text{TPR} = \frac{\text{ROI} \cap \text{Ground Truth}}{\text{Ground Truth}} \quad (4.4)$$

The problem with this metric is that most of the ground truth area is non smoke since ground truth is provided only as a box roughly surrounding the smoke area. Thus maximizing this quantity does not necessarily mean

that we are hitting true smoke regions. Regarding the Jaccard similarity we defined it as follows:

$$Jaccardsimilarity = \frac{ROI \cap Gorund Truth}{ROI \cup Gorund Truth} \quad (4.5)$$

The problem with this metric is that by maximizing it, we can't enforce the fact that we consider hitting the smoke to be more important than reducing the false positive regions.

### 4.3 Experimental Setup

In this section, we aim to find the most relevant features to use, the number of rounds as well as the order by which features are applied. As we previously described in chapter 3, we divided the problem into 2 sub-problems, namely: detecting smoke in forest regions and in sky regions separately. In forest regions, the relevant features are: intensity, diff intensity, diff intensity SD, motion magnitude, motion magnitude SD and correlation. For sky, the relevant features are: diff intensity, diff intensity SD, motion magnitude, motion magnitude SD and correlation.

We decided to tackle each part separately. This is done by splitting each image into 2 parts, one part containing sky superpixels and another with forest superpixels. We also split the ground truth box into 2 parts, one for the smoke in the forest region and one for the smoke in the sky region. We run the pipeline separately and then compute the hit rate and FPR metrics for each part separately. For example for the forest regions, the hit rate is defined as how many smoke regions in all of the forest smoke regions do we hit. While the forest FPR would be the areas of forest out of the forest smoke label that are falsely detected as smoke.

In each part, we determine the best feature configurations that produce best accuracy of smoke detection. Afterwards, We combine best configurations of both parts together and run the experiments but this time calculating the hit rate and the FPR for the whole image instead of for each region separately. We determine the configurations that yield best accuracy in terms of the metrics (high hit rate and low FPR). Finally, we run the the selected configurations on the test set and report the results. We also try the other variant of the algorithm described in chapter 3.7 and compare its results to the main algorithm.

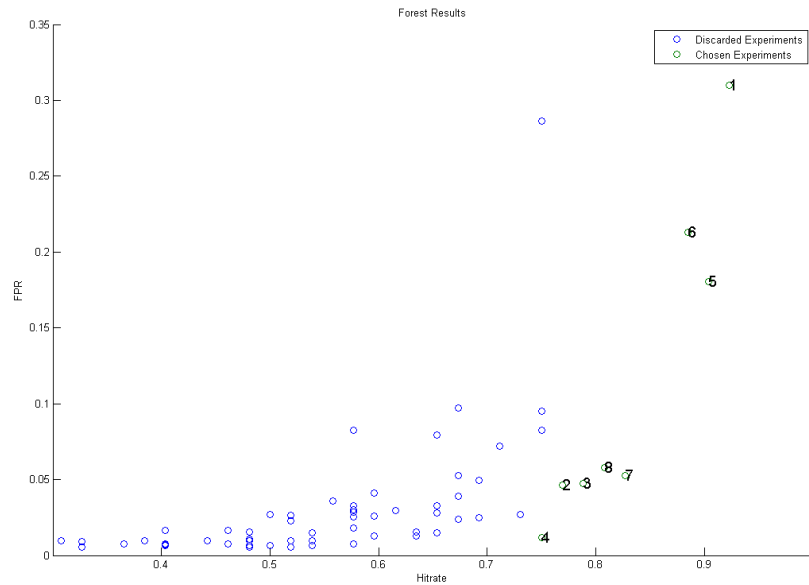
#### 4.3.1 Determining best feature configuration in Forest Region

For the forest region, trying all possible feature combinations with all possible number of rounds meant 1956 experiments ( ${}^6P_1 + {}^6P_2 + \dots + {}^6P_6$ ).

#### 4. EXPERIMENTS AND RESULTS

To reduce the search space, we decided only to first explore the features: intensity, diff intensity, motion magnitude and correlation. Afterwards, we try adding standard deviation features like diff intensity SD and motion magnitude SD as additional rounds in the end. Thus, we initially try all combinations of the four features (intensity, diff intensity, motion magnitude, correlation) with all possible number of rounds (1,2,3 and 4 rounds) resulting in 64 configurations ( ${}^4P_1 + {}^4P_2 + {}^4P_3 + {}^4P_4$ ). We run these configuration and compute for each of them the hit rate and the FPR of the forest part only. We plot the hit rate vs the FPR in order to be able to choose the best configurations to continue experimenting with. Best configurations are those producing a high hit rate while keeping the FPR low. Such configuration are typically found at the bottom right of the plot. We show the results in figure 4.1 with the best configurations plotted in green. We also number the best configurations and provide features used in them in table 4.1.

**Figure 4.1:** Plot of hit rate vs FPR of forest region using all combinations of intensity, diff intensity, motion magnitude and correlation features with all possible number of rounds. Best experiments are plotted in green and configurations shown in table 4.1 .



We choose the green marked experiments (8 configurations) as the best configurations and we discard the rest of the other configurations. Configurations (2,3,4,7,8) achieves high hit rate while maintaining low FPR. We still choose configurations (1,5,6) since they maximize the hitrate and this is our main priority. For the best 8 configurations, we combine them with the

**Table 4.1:** Table showing best forest configurations using features intensity, motion magnitude, correlation, diff intensity

Experiment	Rounds	Features Used
1	1	intensity
2	1	diff intensity
3	2	intensity, motion mag.
4	2	diff intensity, motion mag.
5	2	correlation, diff inetnsity
6	3	correlation, motion mag., diff intensity
7	4	intensity, motion mag., diff intensity, correlation
8	4	motion mag., diff intensity, correlation, intensity

standard deviation features resulting in the following 40 configurations:

- Eight original configurations without any extra rounds
- Eight configurations with an extra round of diff intensity SD
- Eight configurations with an extra round of motion magnitude SD
- Eight configurations with extra two rounds: motion magnitude SD + diff intensity SD
- Eight configurations with extra two rounds: diff intensity SD + motion magnitude SD

We run the 40 configurations and plot the results in figure 4.2. We choose the 10 configurations shown in table 4.2 as the best configurations for the forest part. Kindly note that the ids of the best experiments 1 through 10 are just sequential ids. The first 8 experiments with ids 1 through 8 are different from the first 8 experiments in 4.1 and in table 4.2.

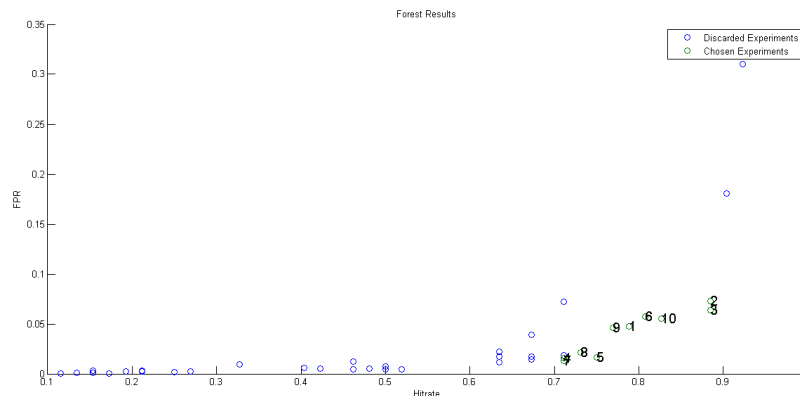
### 4.3.2 Determining best feature configuration in Sky Region

We follow a similar technique in determining the best feature configuration for the sky region. We first pick the features: diff intensity, motion magnitude intensity and correlation, and run all possible configurations with all possible numbers of clustering rounds. This results in 15 ( ${}^3P_1 + {}^3P_2 + {}^3P_3$ ) experiments shown in the figure 4.3 and in table 4.3.

We choose the best 10 configurations that produce the highest hit rates while keeping FPR low and combined them with the motion magnitude standard deviation and the diff intensity standard deviation features like we did in the forest. Results are shown in the figure 4.4 and in 4.4

## 4. EXPERIMENTS AND RESULTS

**Figure 4.2:** Plot of hit rate vs FPR of forest region after adding diff intensity standard deviation and motion magnitude standard deviation features



**Table 4.2:** Table showing best forest configurations after adding motion magnitude SD and diff intensity SD

Experiment	Rounds	Features Used
1	1	diff intensity
2	2	intensity, motion mag. SD
3	2	diff intensity, motion mag. SD
4	3	intensity, motion mag., motion mag. SD
5	2	diff intensity, diff intensity SD
6	5	intensity, motion mag., diff intensity, correlation, diff intensity SD
7	4	intensity, motion mag., motion mag. SD, diff intensity SD
8	4	diff intensity, motion mag., motion mag. SD, diff intensity SD
9	3	diff Intensity, diff Intensity SD, motion mag. SD
10	6	motion Mag., diff Intensity, correlation, intensity, diff Intensity SD, motion Mag. SD

### 4.3.3 Combining forest and sky configurations

We combine the best 10 forest configurations with the best 10 Sky configuration resulting in 100 experiments. We run all experiments and plot the hit rate vs the FPR of the forest and sky parts combined in figure 4.5.

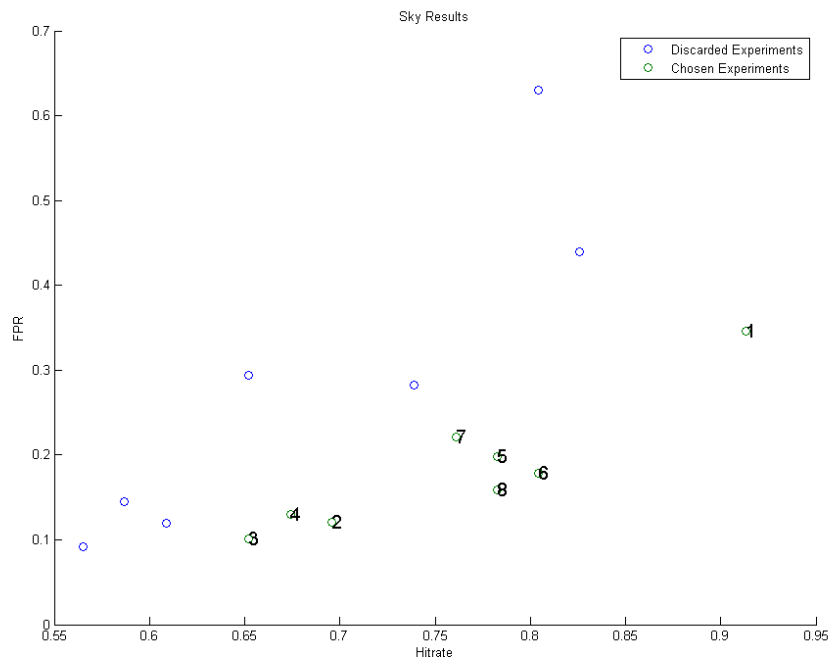
We then select the best 20 configurations and run them on the test set resulting in figure 4.6.

### 4.3. Experimental Setup

**Table 4.3:** Table showing best sky configurations with features diff intensity, motion magnitude, correlation

Experiment	Rounds	Features Used
1	1	diff intensity
2	1	correlation
3	1	motion mag.
4	2	correlation, motion mag.
5	3	diff intensity, correlation, motion magnitude
6	3	diff intensity, motion magnitude, correlation
7	3	correlation, diff intensity, motion mag.
8	3	motion mag., diff intensity, correlation

**Figure 4.3:** Plot of hit rate vs FPR of sky region using all combinations of diff intensity, motion magnitude and correlation features with all possible number of rounds. Best configurations are plotted in green.

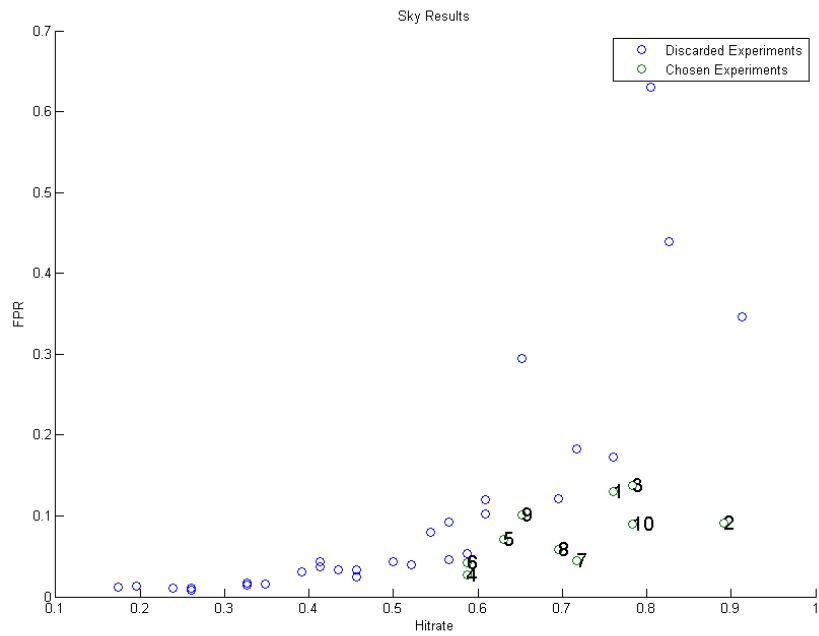


We also tried the variant of performing the clustering step on all superpixels from all triplets at one as described in 3.7. We run the same 20 configurations again on the test set but with clustering all superpixels at once. Results are shown in figure 4.7.

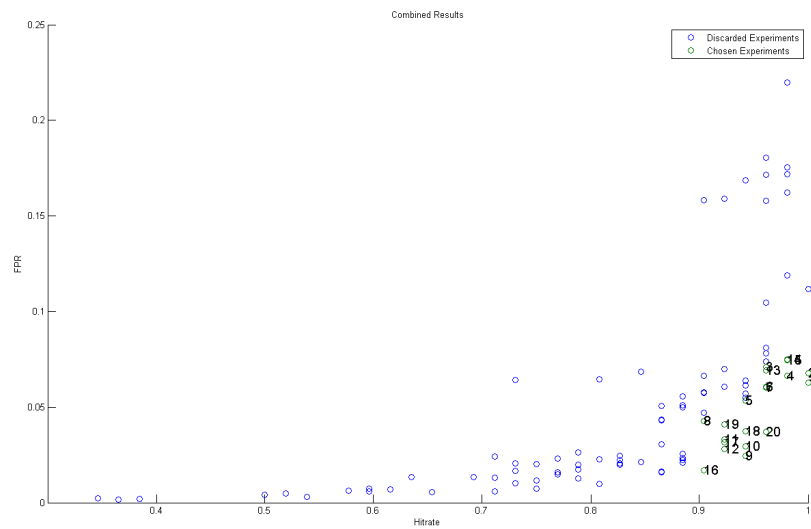


## 4. EXPERIMENTS AND RESULTS

**Figure 4.4:** Plot of hit rate vs FPR of sky region after adding diff intensity standard deviation and motion magnitude standard deviation features. Best configurations are plotted in green.



**Figure 4.5:** Plot of hit rate vs FPR of whole images with forest and sky regions combined together. Best 20 configurations shown in green.

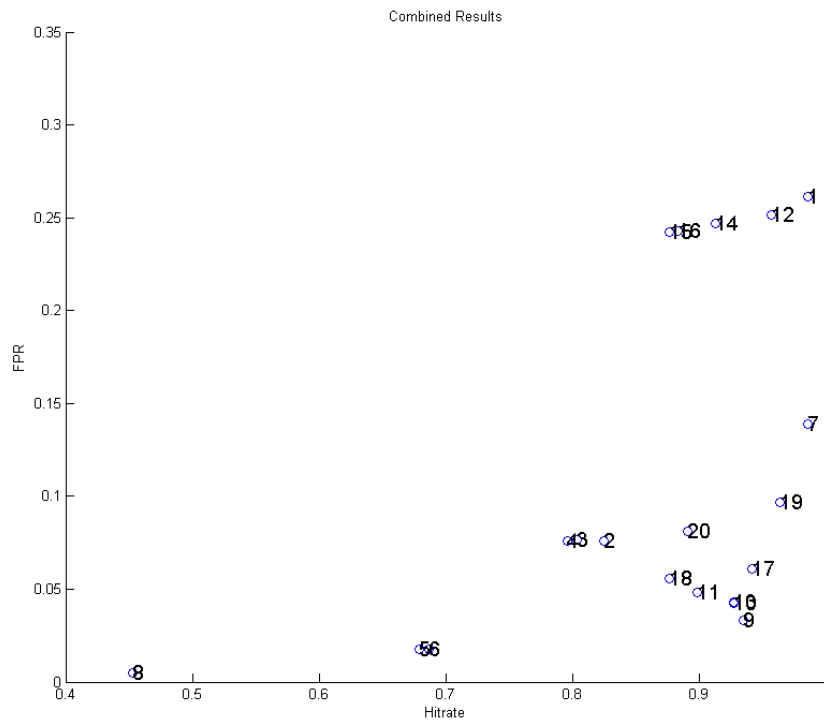


### 4.3. Experimental Setup

**Table 4.4:** Table showing best sky configurations after adding motion magnitude SD and diff intensity SD

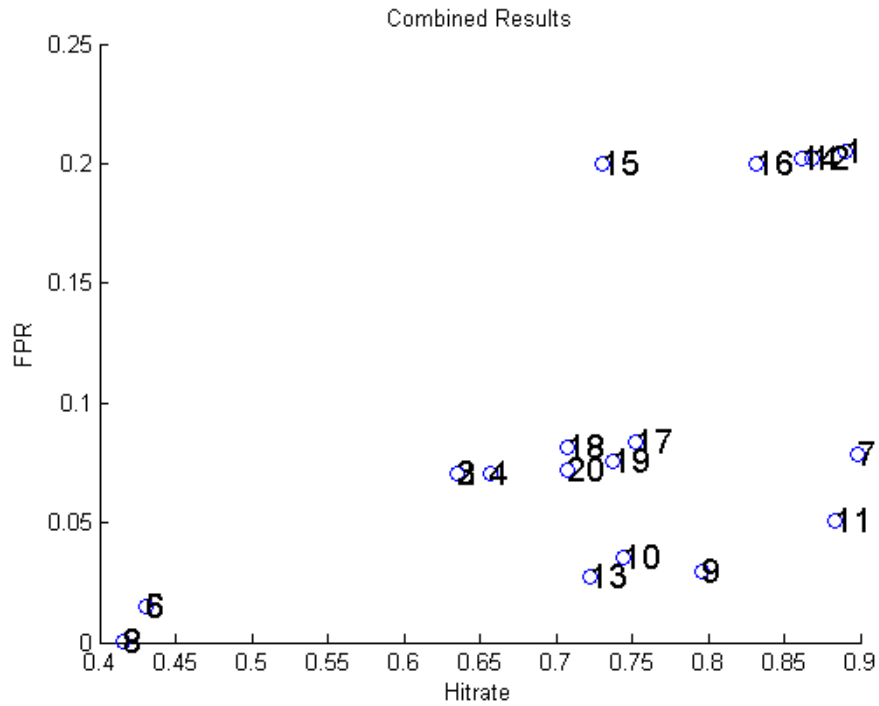
Experiment	Rounds	Features Used
1	1	motion mag.
2	2	diff intensity, motion mag. SD
3	2	motion mag., motion mag. SD
4	2	correlation, diff intensity SD
5	2	motion mag., diff intensity SD
6	3	correlation, motion mag., diff intensity SD
7	3	motion mag., motion mag. SD, diff intensity SD
8	5	diff intensity, correlation, motion mag. motion mag. SD, diff intensity SD
9	5	diff Intensity, correlation, motion mag. diff intensity SD, motion mag. SD
10	5	motion Mag., diff Intensity, correlation diff Intensity SD, motion Mag. SD

**Figure 4.6:** Plot of hit rate vs FPR on the test set



## 4. EXPERIMENTS AND RESULTS

**Figure 4.7:** Plot of hit rate vs FPR on the test set, clustering superpixels from all triplets together at the same time



### 4.4 Visualization Of results

In this section, we show pairs of images showing original smoke image and image with potential smoke regions highlighted. Forest smoke is highlighted in blue while sky smoke is highlighted in green. Results are shown in figures 4.8, 4.9, 4.10, 4.11 and 4.12



**Figure 4.8:** Figure showing algorithm results

### 4.5 Discussion

In this section we try to interpret the results obtained in the previous section. First, we adopted the strategy of searching for the best configurations of for-



**Figure 4.9:** Figure showing algorithm results

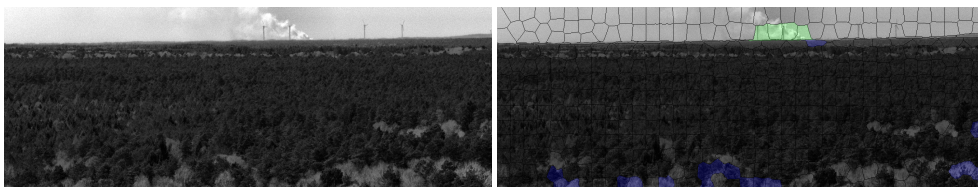


**Figure 4.10:** Figure showing algorithm results



**Figure 4.11:** Figure showing algorithm results. Wind mills are mistakenly included as smoke

est and sky regions separately before combining the configurations. Such a greedy strategy is effective at limiting the search space but can't guarantee reaching the globally optimal combined feature configuration of forest and sky. This is because in a combined feature configuration (forest + sky configuration that solves the problem for the whole image), metrics such as the hit rate relies on the combination of forest and sky configurations. Let's assume that we have two forest configuration  $f_1$  and  $f_2$  such that  $\text{forestHitRate}(f_1) < \text{forestHitRate}(f_2)$ . Let's also assume that there is a sky configuration  $s_1$  and another configuration  $s_2$  such that  $\text{skyHitRate}(s_1) < \text{skyHitRate}(s_2)$ . It can be the case that  $f_1$  when combined with  $s_1$  produce a hit rate higher than



**Figure 4.12:** Figure showing algorithm results. Some forest regions are mistakenly included as smoke

f2 combined with s2. This can occur when the number of distinct triplets in which smoke is hit by combined configuration f1 + s1 is greater than number of distinct triplets in which smoke is hit by the combined configuration f2 + s2.

We can also observe that hit rates achieved in forest part alone 4.2 or sky part alone 4.4 are generally lower than hit rates achieved in combined configurations 4.5. In sky or forest alone, hit rate barely exceeds 90% while in combined configurations we reach perfect hit rates. This is because, when we run experiments on whole images, the hit rate per triplet is one if smoke is hit in sky region or if it is hit in forest region or in both.

In general, FPR is lesser in case of forest experiments 4.2 compared to forest experiments (figure 4.4) with most of the forest experiments achieving a FPR of less than 0.1 compared to sky experiments where FPR is generally lower than 0.2. We believe that this is caused by the fact that sky region has more noise than the forest region. We believe that photon noise follows a Poisson distribution, which means that the standard deviation is equal to the mean. The more photons that hit the camera, the higher the noise. This is the case with sky regions since they are usually brighter.

When applying the algorithm on test set (figure 4.6), we notice a very good performance. Configurations 9 and 10 achieve hit rates of 94% and 93% respectively while having FPR of 0.035 and 0.04 respectively.

When comparing the main algorithm results on the test set 4.6 to the algorithm's variant results on the test set (figure 4.7), we notice that our algorithm performs better in terms of hit rate when considering one image at a time. This is shown in configurations 1 and 7 that achieve 97% hit rate when algorithm is applied on a single image at a time compared to when all superpixels are clustered together (both achieving less than 90% hit rate). This can be explained by the fact that feature distributions vary across different images. For example, We consider two images, one with a very dark forest and the second with a very bright forest. The intensity values of the darker forest would be lesser than intensity values of a bright forest. This means that brighter parts of the darker forest wouldn't have same intensity values as brighter parts of bright forest. Thus combining superpixels of both forests together and clustering on intensity feature and choosing the bright superpixels from the set of combined superpixels would most probably lead to discarding the bright regions of the darker forest and leading to potentially missing the smoke in it. We also observe that considering one triplet at a time as in the main algorithm cause a higher FPR than when all images are clustered at once. However, we give higher priority to the hit rate than the false positive rate since missing smoke is more dangerous than having extra false positive regions.

It should be noted that for the best performing configurations (configura-

tions 9,10) on the test set 4.6, the forest features used are: diff intensity followed by motion magnitude SD. For the sky part, configuration 9 uses motionMagnitude, motionMagnitude SD, diff Intensity SD while configuration 10 uses: diff Intensity,correlation,motion magnitude,motion magnitude SD,diff intensity SD. We think this is the case in the forest as diff intensity helps choosing the moving parts while the motion magnitude sd separates the smoke from the rest of the moving parts. For the sky part, the diff image intensity isn't as good due to the noise present in the sky. However, the sequence of features motion magnitude, motion magnitude SD, diff intensity SD can be explained in a similar way. The motion magnitude first chooses the moving parts while the motion magnitude SD and diff intensity SD help separating smoke from non-smoke regions.



# Conclusion

---

In this thesis, we developed an algorithm for smoke region detection. The algorithm takes as input triplets of images of the same scene (though can theoretically take image sequences of any length) and detects potential regions of smoke in a semi-supervised manner. We separated the problem into two subproblems, namely: detecting smoke regions in forest and sky regions separately. We investigated the use of several features for each of the subproblems and experimented various ways of combining them to achieve the best accuracy of smoke detection. At each round, we pick a new feature and use it to discard non-smoke regions. The algorithm is relatively simple to understand since we only use one feature at each round and rely on the simple yet powerful Expectation Maximization algorithm to differentiate between potential smoke and non-smoke regions. This makes it easier to debug the pipeline, incorporate new features and adapt it to new scenes. All what is required is to retrain the model using new scenes, select the best features to use in the new conditions and incorporate newer features if necessary.

We tested the algorithm on a test set captured later in time and contained new scenes. We could achieve a 94% hit rate while keeping the FPR at 0.04 (configuration 9 of the test set using diff intensity, motion magnitude SD for forest and diff intensity, motion magnitude SD and diff intensity SD for the sky).

### 5.1 Model limitations and Future work

The main model limitation is that it only uses local information of superpixels. This means that a superpixel with similar appearance as smoke and some motion can be mistakenly detected as a potential smoke region as in figure 5.1.



This limitation can be addressed by using Markov Random fields to incorporate global information. For example, a superpixel is more likely to be a smoke superpixel if its neighbour is a smoke superpixel too.

The pipeline presented in this thesis can be extended by incorporating more features. Some of these features include:

- Texture description features: It can be useful to add features that describe smoke patterns, An example of such a feature is the Local Binary Pattern [4].
- Wavelet features: Smoke tends to blur the region behind it leading to a reduction in the high frequency components of the image. Wavelets can help determine the drop in the high frequency components in a particular region

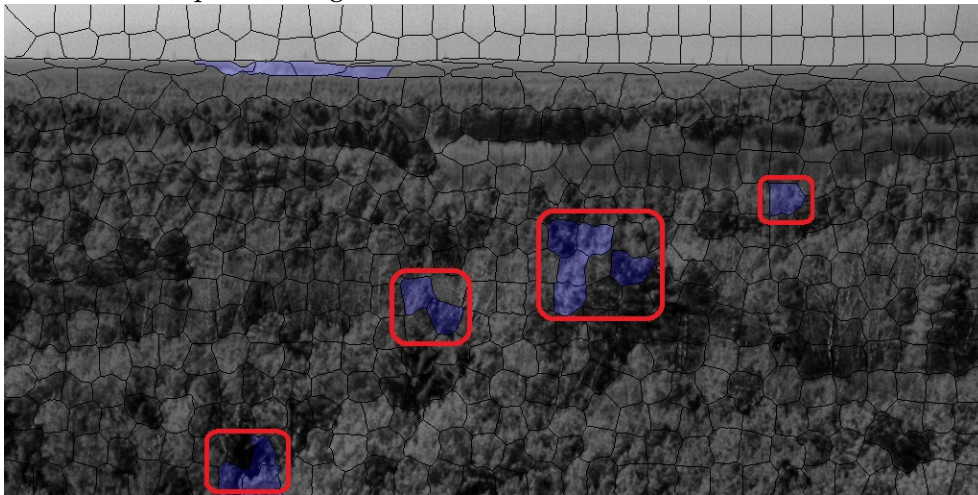
Another Idea is to employ object detection techniques to remove certain objects from the images. An example of this is implementing a windmill detector that removes superpixels containing windmills/electricity towers from the sky. This can be achieved by using a vertical edge detector operator to determine such objects.

Last but not least, by manually labelling the exact smoke regions, one can use the whole domain of supervised learning to be able to solve the problem. Manual labels can be obtained via crowd-sourcing or through the help of volunteers. After extracting the regions of interest using the algorithm described in this thesis, extracted regions can be used to train a classifier.

Original Image



False positive regions similar to smoke surrounded in red



**Figure 5.1:** False positive Smoke similar superpixels



---

## Bibliography

---

- [1] R. C. Conzalez and R. E. Woods, *Digial Image Processing*. New Jersey: Prentice-Hall, Inc., 2001.
- [2] S. M. Pizer, E. P. Amburn, J. D. Austin, et al.: *Adaptive Histogram Equalization and Its Variations*. Computer Vision, Graphics, and Image Processing 39 (1987) 355-368.
- [3] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Susstrunk, *SLIC Superpixels*. EPFL Technical Report 149300, June 2010.
- [4] H. Tian; W. Li; P. Ogunbona ; D. T. Nguyen; C. Zhan, *Smoke detection in videos using Non-Redundant Local Binary Pattern-based features*. Multimedia Signal Processing (MMSP), 2011 IEEE 13th International Workshop, pp.1-4, 17-19 Oct. 2011
- [5] J. Kwak, B. C. Ko. J. Nam, *Forest Smoke Detection Using CCD Camera and Spatial-temporal Variation of Smoke Visual Patterns*. Computer Graphics, Imaging and Visualization (CGIV), 2011 Eighth International Conference, pp.141-144, 17-19 Aug. 2011
- [6] S. Wang, Y. He, J. J. Zou, D. Zhou, J. Wang, *Early smoke detection in video using swaying and diffusion feature*. Journal of Intelligent & Fuzzy Systems, pp. 267–275, 2014
- [7] F. Yuan, *A fast accumulative motion orientation model based on integral image for video smoke detection*. Pattern Recogn. Lett. 29, 7 , pp. 925-932, May 2008.
- [8] *Cambridge Advanced Learner's Dictionary (Third ed.)*. Cambridge University Press. 2008. ISBN 978-0-521-85804-5.

- [9] National Interagency Fire Center, *Total Wildland Fires and Acres (1960-2009)*. URL: [http://www.nifc.gov/fireInfo/fireInfo\\_statstotalFires.html](http://www.nifc.gov/fireInfo/fireInfo_statstotalFires.html) (Retrieved 13/10/2015).
- [10] O. Sokrine, *Least-Squares Rigid Motion Using SVD*. URL: [https://igl.ethz.ch/projects/ARAP/svd\\_rot.pdf](https://igl.ethz.ch/projects/ARAP/svd_rot.pdf) (Retrieved 10/10/2015).
- [11] Ahmad A. A. Alkhatib, *A Review on Forest Fire Detection Techniques*. International Journal of Distributed Sensor Networks, vol. 2014, Article ID 597368, 12 pages, 2014.
- [12] R. Sridharan, *Gaussian mixture models and the EM algorithm*. URL: <http://people.csail.mit.edu/rameshvs/content/gmm-em.pdf> (Retrieved 1/10/2015)
- [13] C. Liu, *Beyond Pixels: Exploring New Representations and Applications for Motion Analysis*. Doctoral Thesis, Massachusetts Institute of Technology, May 2009.
- [14] Wikipedia, *Optical Flow*, URL: [https://en.wikipedia.org/wiki/Optical\\_flow](https://en.wikipedia.org/wiki/Optical_flow) (Retrieved 2/10/2015)
- [15] N. Brovko, R. Bogush, S. Ablameyko, *Smoke detection algorithm for intelligent video surveillance system*. The Computer Science Journal of Moldova 21 , no. 1, pp 142-156, 2013.
- [16] T. X. Truong, J. Kim, *An early smoke detection system based on motion estimation*. Strategic Technology (IFOST), 2010 International Forum, pp.437-440, 13-15 Oct. 2010.



## Declaration of originality

The signed declaration of originality is a component of every semester paper, Bachelor's thesis, Master's thesis and any other degree paper undertaken during the course of studies, including the respective electronic versions.

Lecturers may also require a declaration of originality for other written papers compiled for their courses.

I hereby confirm that I am the sole author of the written work here enclosed and that I have compiled it in my own words. Parts excepted are corrections of form and content by the supervisor.

**Title of work** (in block letters):

Machine Learning Approach for Adaptive Automated Smoke Region Detection

**Authored by** (in block letters):

*For papers written by groups the names of all authors are required.*

**Name(s):**

Ahmed

**First name(s):**

Amr

With my signature I confirm that

- I have committed none of the forms of plagiarism described in the 'Citation etiquette' information sheet.
- I have documented all methods, data and processes truthfully.
- I have not manipulated any data.
- I have mentioned all persons who were significant facilitators of the work.

I am aware that the work may be screened electronically for plagiarism.

**Place, date**

Zurich, 20/10/2015

**Signature(s)**

Amr Ahmed

*For papers written by groups the names of all authors are required. Their signatures collectively guarantee the entire content of the written paper.*