Extracting Effective Visual Words From Images For Topic Modelling

M.Tech Mini Project Report
Submitted in Partial Fulfillment of the Requirements for the Degree of

M. TECH. System-on-Chip Design (SOCD)

By

Anupurba Mitra(152002003)

Under the guidance of,
Dr. Mrinal Kanti Das
Assistant Professor
Computer Science And Engineering

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING INDIAN INSTITUTE OF TECHNOLOGY PALAKKAD

CERTIFICATE

This is to certify that the work contained in this project entitled ***"Extracting Effective Visual Words From Images For Topic Modelling"*** *is a bonafide work of* ***Anupurba Mitra(Roll No. 152002003)****, carried out in the Department of Computer Science and Engineering, Indian Institute of Technology Palakkad under my supervision and that it has not been submitted elsewhere for evaluation*

Dr. Mrinal Kanti Das
Assistant Professor
Department of Computer Science and Engineering
Indian Institute of Technology, Palakkad

Contents

List of Figures

Abstract *We propose a novel approach towards generating effective visual words from images in this mini project. The proposed task requires images to be segmented first, using Gaussian Mixture Model (that is, the histogram of the pixel intensities of an image fit to a mixture of Gaussian's), followed by hand crafted feature extraction using SIFT on the segmented images. Then we perform K means clustering on the feature space, to generate visual words. In order to evaluate the effectiveness of the visual words we apply it to a LDA based topic model. Even though it is more common to directly apply the visual words in other CV tasks like image classification, compression , texture recognition etc, we have applied the topic modelling results for image classification (in both a supervised setting and an unsupervised setting). The objective of our task was : first, to generate effective visual words, and second, to evaluate the effectiveness of those visual words in various domains of CV.*

The code for this mini project is available here The google colab notebook is divided into sections. All the matplotlib visualizations displayed in the report, present in the section **"All matplotlib visualizations used in the project report"** . Also code and visualization of the performance of the naive , supervised and unsupervised classification approaches are given in the sections : **"Supervised classification of images into burst and non burst : Naiive Approach" , "Supervised Image Classification Technique; Based on topic modelling results", "Unsupervised classification based on topic modelling results"** respectively.

The intermediate image segments generated and the LDA results can be loaded from here

1 *Introduction*

As our collective knowledge continues to be digitized and stored, in the form of blogs, web pages, scientific content and visual information, it becomes more and more difficult to organise, search and understand this vast amount of information. Biologically, one third of the cortical area of human brain is involved in visual information processing. Thus, it is obvious that the processing of visual information and making them more interpretable for a machine has garnered the attention of AI enthusiasts for decades now. It has been observed that many algorithms that are aimed at making images (that are actually represented as an array of pixels) more interpretable for the computer, have actually been motivated by their text - only analogues, like tf-idf (Salton and McGill, 1983) or bag of words or continuous bag of words [21].

.

In the proposed work, we are implementing the Bag of Visual Words algorithm (BoVW). Several works have been discussed in [10, 6, 9] for improving the BoVW algorithm. Some deep learning based approaches, have been discussed in [16, 8] and also, some approaches were discussed, where in a generative machine learning models were trained to learn visual representations from images, such that if provided with an augmented or distorted version of the same or similar images, they could generate the appropriate visual words [12, 11]. [22] also discussed, how we can generate descriptive visual words, such that one visual word actually represents a group of similar pixels (i.e. pixels with identical semantic meaning).

But, we are implementing a novel technique where, the feature extraction is implemented on image segments and not directly on images , but on image segments, with the objective of making them more interpretable for learning features from images in an unsupervised setting.

1.1 Problem Statement

The main tasks defined for this project are enlisted below,

1. First task was to implement the BoVW algorithm using image feature extractors. We have applied different feature extraction techniques on some coloured images, and finally decided to go forward with SIFT feature extraction.
2. The BoVW was to be used for classification of the images in an unsupervised manner.
3. Secondly, the BoVW was to be formed for some standard datasets, and the results were to be fed as input a LDA based topic model.
4. The effectiveness of the resulting BoVW was to be analyzed. Here, We have applied the topic modelling results for classification of the images in both supervised and unsupervised settings, for evaluating the effectiveness of the visual words in the context of topic modelling.

1.2 Data Description

The data set provided for the aforementioned tasks was "Burst 50 " data set, which is a collection of 100 coloured images (50 belonging to burst category and 50 belonging to non burst category), of dimensions 1273 X 833 X 3. Here 3 denotes the 3 color channels (RGB).

1.3 Tools and Technologies used

Programming Languages	Python
Python Libraries	Numpy, Pandas, Matplotlib, sklearn [5], scikit-image, cv2
Other tools	Google Colab

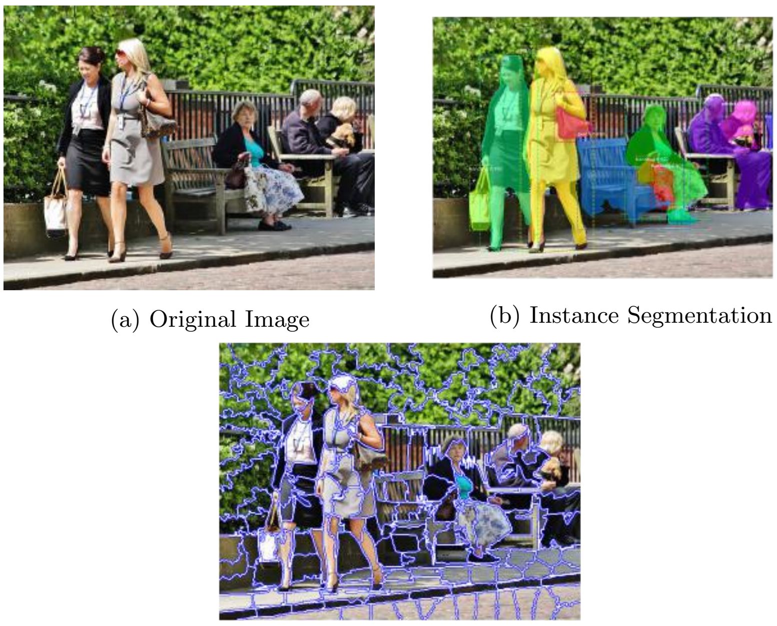
The rest of the report is organised as follows ; section 2 describes the methodology used and discusses different results. In sections 3 I have concluded the report and discussed the future scope of this project work.

2 *Methodology*

2.1 Image Segmentation using Gaussian Mixture Models

Image Segmentation is a Computer Vision (CV) task, that aims at partitioning the images into some parts, called ”segments”, such that they are more interpretable for a computer system as well as consume less space. Applications of image segmentation include object detection, texture recognition, image classification, etc. There are mainly 3 types of image segmentation techniques, listed as follows:

- **Superpixel based image segmentation** , that is grouping of image pixels into small collections of pixels based on color, texture and other low level primitives. Popular algorithms include SLIC [1] (Simple Linear Iterative Clustering).
- **Semantic segmentation :** This technique aims segregating the images such that each pixel is associated with a label and can be identified distinctively, but different parts/components of the image cannot be identified separately.
- **Instance based segmentation :** Here, instances of the same class of objects within an image are uniquely identified by bounding boxes, but each and every pixel of the image is not labelled. The aim is more to find specific objects from the image. A popular deep learning based approach for this purpose is the Mask RCNN [14] model, results of application of which are shown in figure 1 (b).



(a) Original Image

(b) Instance Segmentation

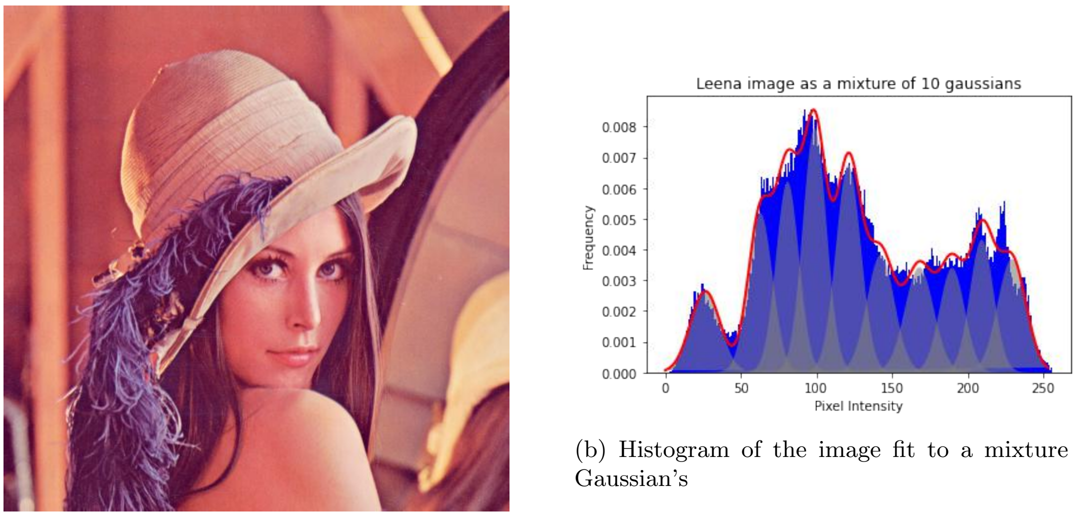
(c) Super Pixel Based Segmentation using SLIC

Figure 1: Illustration of Different image segmentation techniques

- **Panoptic segmentation [15]:** It is a combination of both instance segmentation and semantic segmentation. That is, every pixel of the image is labelled, and labels belonging to different instances of the same class are uniquely identified.

The above algorithms are more suitable for object detection and other CV tasks based on Supervised learning. But, in a supervised setting, we need annotated images, and it is often a mammoth task to manually annotate images. Thus, our objective was to make images more interpretable for unsupervised learning and topic modelling tasks. Therefore, we have used Gaussian Mixture Model and the Expectation Maximization algorithm to segment images, such that one segment contains only one component of the image.

Gaussian Mixture Models is a probabilistic model that assumes, data to be composed of a mixture of a finite number of Gaussian Distributions, with unknown parameters.The Gaussian Mixture Model actually implements the Expectation Maximization algorithm, for fitting the data to a mixture of Gaussian's. It is mainly used for grouping similar types of records into a single cluster, and thus is widely used as an unsupervised Machine Learning (ML) algorithm. Figure 2 shows the histogram of an in image as a mixture of Gaussian's.



(b) Histogram of the image fit to a mixture of 10 Gaussian's

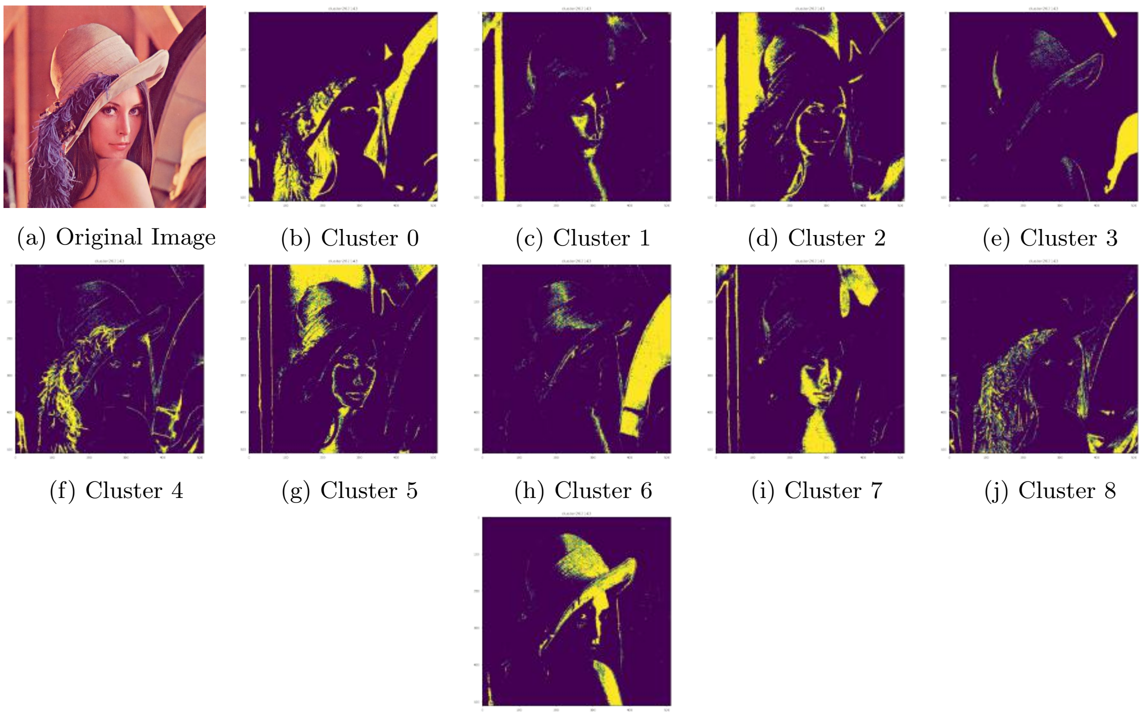
(a) Original Image

Figure 2: Image Segmentation using GMM

Some images and their segments are shown in figure 3.

The reason of using Gaussian Mixture Model for image segmentation, over the more widely used technique of K means clustering, is owing to its ability to perform soft assignment of data points to clusters. Although both GMM and K means clustering work on similar principles, and even the computational time of K means is relatively less, it assigns one data point to one and only one cluster (that is hard assignment). However, in real world data, we may not have well defined boundaries between data points. Rather it is more obvious to have data points which belong to more than one cluster with some probabilities (that is soft assignment).Thus, it is more helpful in grouping the semantically identical pixels into one cluster.In the context of topic modelling, it is crucial, as preserving the semantic meaning in between data points is the very basis of topic modelling. GMM can also give oblong clusters, unlike K means which can only produce circular clusters.

The optimal number of Gaussian's into which the data must be grouped is given by the "Elbow method", using the ACI (that is, Akaike information

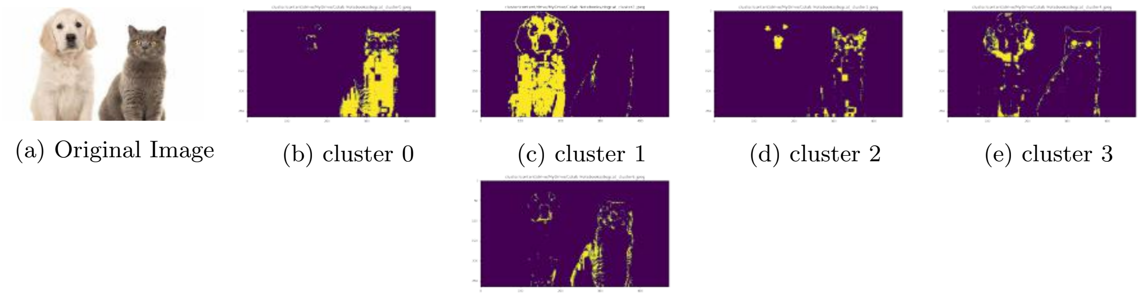


(a) Original Image (b) Cluster 0 (c) Cluster 1 (d) Cluster 2 (e) Cluster 3

(f) Cluster 4 (g) Cluster 5 (h) Cluster 6 (i) Cluster 7 (j) Cluster 8

(k) Cluster 9

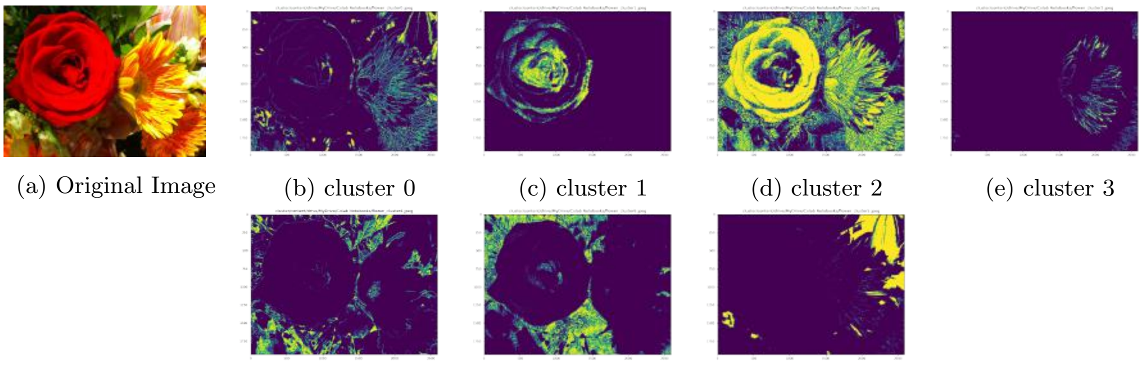
Figure 3: Segments of the original image ”Lenna_(test_image).png”



(a) Original Image (b) cluster 0 (c) cluster 1 (d) cluster 2 (e) cluster 3

(f) cluster 4

Figure 4: Segments of image ”dog_cat.jpg”

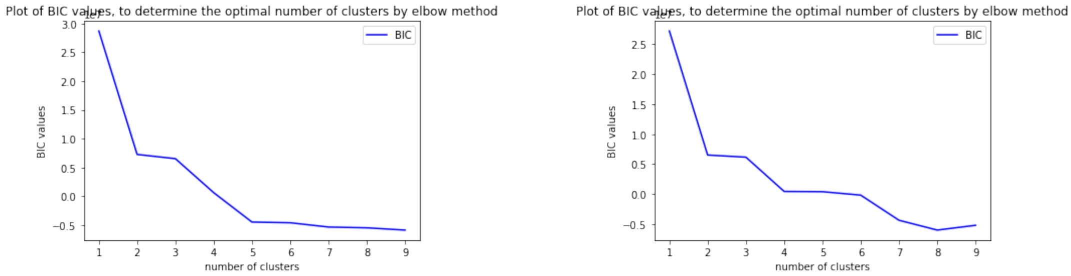


(a) Original Image (b) cluster 0 (c) cluster 1 (d) cluster 2 (e) cluster 3

(f) cluster 4 (g) cluster 5 (h) cluster 6

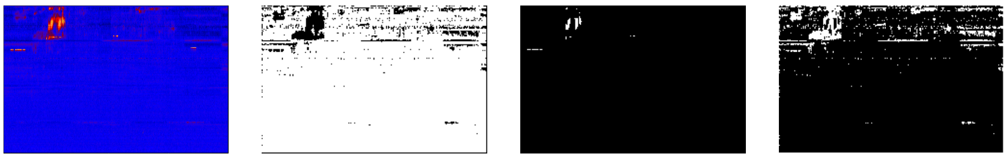
Figure 5: Segments of image ”rose-daisy-flower.jpg”

criterion) and BCI (that is, Bayesian information criterion) scores. The elbow method is as follows; We plot the number of clusters, n (x axis) vs AIC or BIC score corresponding to n (y axis).The x - coordinate where the first elbow occurs, is taken as the optimal number of clusters.Here, for the burst images that we have used (which had pixels showing noise, burst and foreground), the first elbow was obtained at x = 3, for the non burst images (which only had pixels showing noise and foreground), the first elbow was obtained at x = 2 . The Plots in figure 6 show the BIC score curve of both burst and non burst images and subsequently, we show segments of one of these images belonging to each class (randomly picked) in figure 7 and 8.



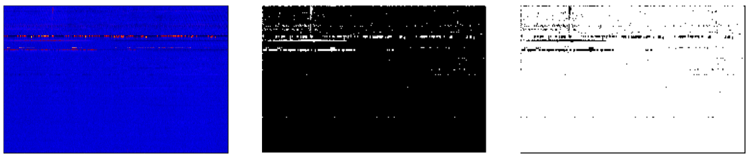
(a) BIC score curve for the class of Burst image (b) BIC score curve for the class of Burst image

Figure 6: BIC vs #clusters curve for burst and non burst images



(a) Original Image (b) Cluster 0 (c) Cluster 1 (d) Cluster 2

Figure 7: Segments of a randomly picked Burst image



(a) Original Image (b) Cluster 0 (c) Cluster 1

Figure 8: Segments of a randomly picked Non-Burst image

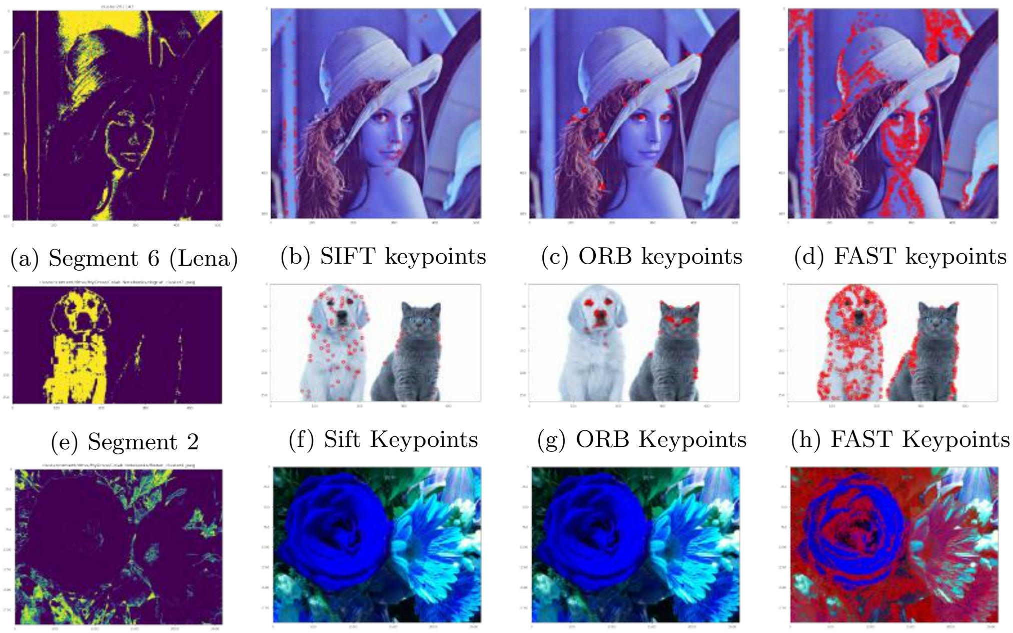
2.2 Feature Extraction

In the next step, we will apply Feature Extraction algorithms on the image segments extracted using GMM, and implement the bag of visual words algorithm on these feature descriptors. We have applied some famous algorithms, namely SIFT (Scale Invariant Feature Transform),FAST (Features from Accelerated Segment Test) and ORB (Oriented FAST). Objective being, just as in a text documents, a ”word” helps us in mapping the documents to a specific collection of topics, ”visual words” must be interpretable enough to map images in a dataset to a collection of topics. The BoVW should also help in mapping the objects of an image to their respective classes (line car, bus, flower, etc).

The figures 9 show the key points extracted from application of SIFT, ORB and FAST feature extraction algorithms to the images whose segments were shown in the previous sub section.

The SIFT algorithm, proposed by Lowe is invariant to, affine transformations, intensity changes, and viewpoint change in matching the image features. Whereas, FAST [19] does not compute the orientation (view point) and is rotation variant. ORB [20] is a fusion of FAST and BRIEF feature extraction algorithms. For the given dataset, we decided to go with SIFT, however, in the future ORB, FAST and BRIEF algorithms can also be applied to the same and a comparative study can be facilitated.

In the next subsection, we will describe the SIFT feature extraction algorithm in detail.



(a) Segment 6 (Lena) (b) SIFT keypoints (c) ORB keypoints (d) FAST keypoints

(e) Segment 2 (f) Sift Keypoints (g) ORB Keypoints (h) FAST Keypoints

(i) Segment 4 (j) SIFT keypoints (k) ORB keypoints (l) FAST keypoints

Figure 9: Keypoints obtained from the segmented images and plotted on the original image for proper visualization. (using SIFT, ORB and FAST feature extraction techniques)

2.2.1 SIFT Feature Extraction

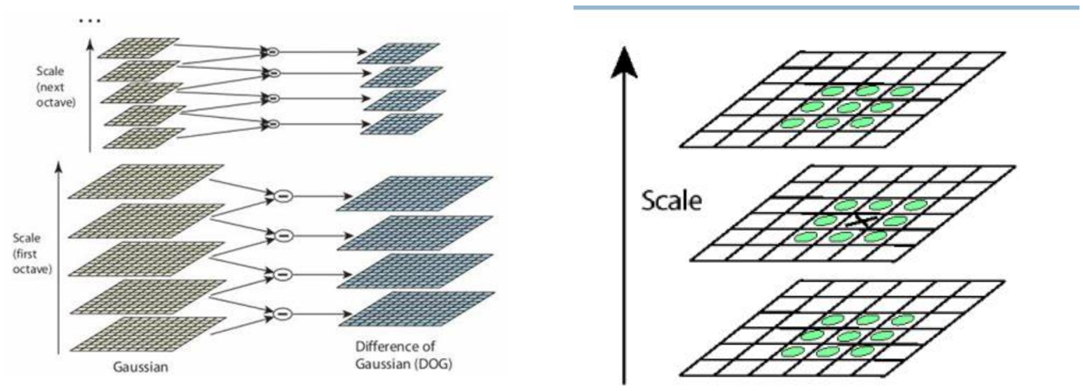
SIFT [17] helps locate the local features in an image, commonly known as the ”keypoints” of the image. These keypoints are scale rotation invariant. Each keypoint is identified by a unique vector of length 128 floating point numbers known as ”Keypoint descriptors”. The process can be divided into four steps:

1. **Constructing a Scale Space :** Scale space is a collection of images having different scales, that are generated from a single image. Distinct features from the image are obtained, ignoring noise. In order to reduce noise from the image, Gaussian Blurring is applied a number of times. This is an iterative step, where, first the image is scaled down to a particular sized image (Octave), followed by application of Gaussian Blurring several time. Ideally, the number of octaves should be 4, and number of times Gaussian Blurring is applied to each octave should be 5.

2. **Difference of Gaussian (DOG) :** The DOG is a feature enhancement technique that involves subtraction of one blurred version of an image from another less blurred version. For each octave, DOG creates a set of images by subtracting the next image pixel intensity values from the previous one. The visual representation of DOG is shown in figure 10(a).

3. **Keypoint Localisation :** This step is further divided into two steps:

   (a) Find the local maxima and minima : To locate the local maxima and minima, every pixel of the image is compared to its neighboring pixels. Here, neighboring pixels includes the 8 pixels surrounding a specific pixel as well as the 9 pixels from the previous and next octave, that is, every pixel value is compared with 26 other pixel values and it is determined whether the said pixel is a local maxima or a local minima. The 26 pixels with which a said pixel is compared, are shown in figure 10 (b)



(a) Difference of Gaussian Feature Enhancement technique

(b) Finding the local maxima and minima

Figure 10: Intermediate steps of SIFT feature Extraction Algorithm (Image has been borrowed from the original research paper [17])

(b) Remove low contrast keypoints (keypoint selection) : Some of the keypoints generated from the above steps may not be robust to noise. Thus, in this steps, those keypoints which have low contrast or lie very close to the edge, are eliminated.

4. **Orientation Assignment :** For each pixel, the orientation and magnitude are calculated by obtaining the gradients in the x and y directions. The orientation is given by :

$$\theta(x,y) = \tan^{-1}(G_y/G_x) =$$
$$\tan^{-1}(G(x,y+1) - G(x,y-1)/G(x+1,y) - G(x-1,y))$$

The magnitude is given by :

$$\text{m(x,y)} = \sqrt{G_x^2 + G_y^2} =$$
$$\sqrt{(G(x,y+1) - G(x,y-1))^2 + (G(x+1,y) - G(x-1,y))^2}$$

5. **Keypoint Descriptor :** Now we have the set of stable keypoints, that are scale and rotation invariant. In this step, for each pixel, the orientation and magnitude of its neighboring pixels are used to generate a unique fingerprint (of length 128) for each pixel, known as the feature descriptor. Since the surrounding pixel's information is used, these descriptors are independent of illumination or brightness of the image.

   First, a 16x16 image patch corresponding to the neighborhood of a pixel is considered, then, this 16x16 block is further sub divided into 4x4 sub blocks, and 8 bins (magnitude and orientation values) are considered out of the 64 bins of each sub block. Thus, the length of each feature descriptor is given as :

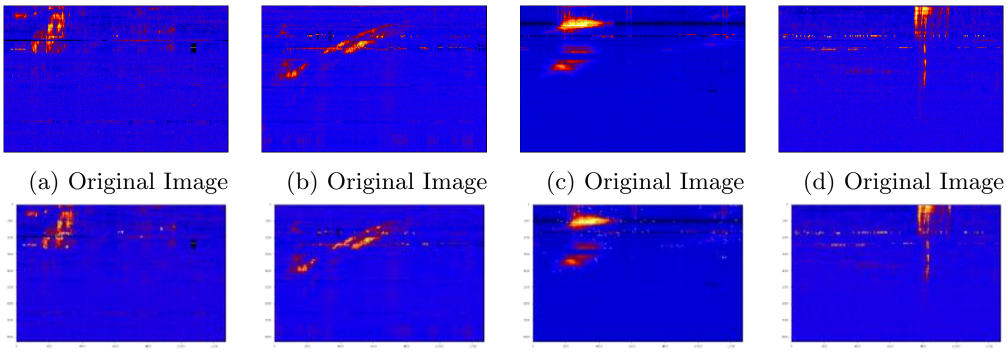
$$\frac{16\times16}{4\times4} \times 8 = 128 \text{ bins for each descriptor}$$

Some randomly picked burst images and their SIFT keypoints mapped to them are are shown in figure 11.

2.3 BoVW Implementation

2.3.1 Traditional BoVW

The traditional Bag Of Visual Words algorithm is an approach that offers mid-level descriptors (MLD) which help in reduction of the semantic gap



(a) Original Image (b) Original Image (c) Original Image (d) Original Image

(e) SIFT keypoints (f) SIFT Keypoints (g) SIFT Keypoints (h) SIFT Keypoints

Figure 11: SIFT features of some randomly extracted burst images

between High Level Descriptors (HLD) and Low Level Descriptors (LLD), that is pixel specific information. It is performed in the following steps:

- Extraction of local feature descriptors using feature extraction algorithms.
- Learning of a ”Visual Codebook”.
- Creating mid level representation of the image using the visual codebook. As for instance, Histogram showing the frequency of visual words in the image dataset is one such mid level image representation

But, the BoVW algorithm suffers from several disadvantages. Firstly, Semantic Gap is the most undesirable effect of BoVW. Actually, in traditional BoVW, feature extraction, codebook generation and formulation of Mid Level Representations, are performed as 3 independent steps. Thus, the semantic relationship in between High level features is not captured by the mid level representation. Again, since the algorithm ignores the ordering of pixels, semantic relationship between low level pixel based information is also ignored.

Secondly, the choice of feature extraction algorithm plays an important role. We must choose an extractor, which is independent to scale and orientation of the image, otherwise, viewpoint changes may make previously unseen parts of the image visible, or can even obstruct or hide parts of the image, which will again affect the performance of the BoVW.

Lastly, BoVW assumes that all the visual words are ”Equally - Likely” to occur in the image, but in reality, some visual words may occur more frequently as compared to others.

We have attempted to overcome some of the above disadvantages. We have chosen SIFT as the feature extractor, which is independent of the scale and orientation of the image. In order to overcome the issue of ”Equally - Likely” occurrence of visual words in the images, we have applied topic modelling on the visual words, which associates a probability of occurrence of each image in a class of images or in an ”Image Topic”, with each visual word and also claims to capture the semantic meaning between low level pixel information and high level features.

2.3.2 Our approach

A total of 7621 SIFT feature descriptors were obtained from the input images (burst and non burst together). Now, for implementing the Bag of Visual

Words algorithm, we applied K means clustering to the feature descriptors generated by SIFT feature extraction.

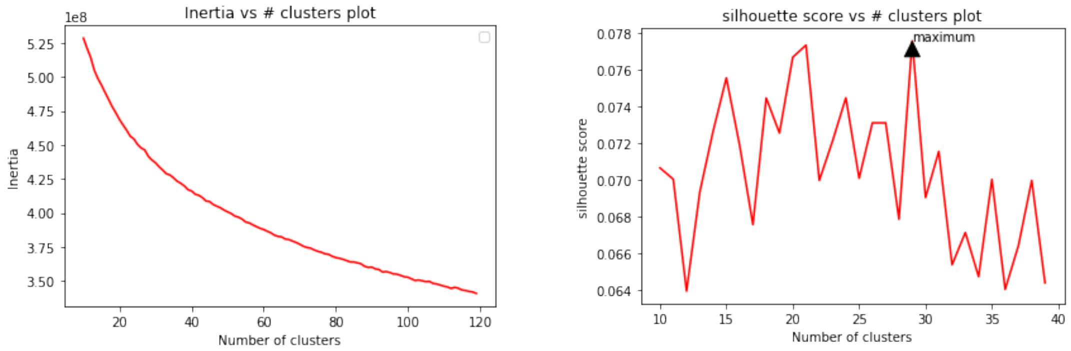
The optimum number of clusters for the input data (7621 feature descriptors) was obtained at 30, using the ”silhouette score” analysis of K means, which peaks at number of clusters = 30. The silhouette score is calculated as follows :

1. calculate the average distance of all the data points from the specified data point in the same cluster (a_i).
2. calculate the average distance of all the data points from the specified data point in the closest cluster (b_i)
3. calculate the coefficient, as follows :

$$\frac{b_i - a_i}{max(a_i, b_i)}$$

The elbow method, which depends on the inertia values of the data points was also implemented for obtaining the optimum number of clusters.Inertia actually is the sum of Euclidean distance of all the points within a cluster from the but, the graph showed a smooth descent and it was difficult to figure out where the elbow has actually occurred.

Figure 12 (a & b) shows the plot of silhouette score vs optimum number of clusters and inertia vs optimum number of clusters plot for the said input data.



(a) Inertia vs number of clusters

(b) Silhouette Score vs number of clusters

Figure 12: Estimation of optimal number of clusters for K means

The result of the clustering is a ”visual codebook” with 30 visual words.The visual words are actually cluster centroids that have been obtained as a result of K means clustering on the 7621 SIFT feature descriptors. Each cluster centroid is identified by a index value ranging from 0 to 29. After the clustering, the SIFT feature space is quantized (by mapping the feature descriptors to the visual code book). The histogram in figure 13 shows the top 10 most frequently occurring visual words.

2.4 Application of BoVW on LDA based Topic Model

Probabilistic models for inferring hidden topics from documents are one of the greatest success stories of unsupervised learning. In generative probabilistic modelling, data(corpora) is treated as being modelled by a generative procedure, that considers hidden variables(also called latent variables). Now, as humans, we understand the semantics of such text documents and the

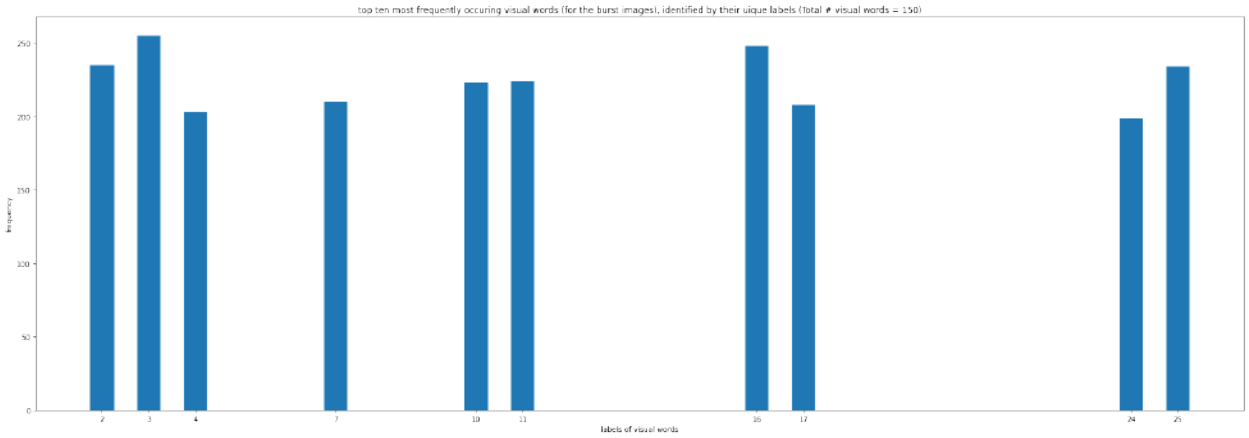


Figure 13: Top 10 most frequently occurring visual words (2,3,4,7,10,11,16,17,24,25)

meanings of words, but to make a computer(that understands only 0s and 1s), is a difficult task. Ground breaking work has been done in this field, by David M. Blei [4], who formulated Latent Dirichlet allocation(LDA). Inspired by the recent improvements in topic modelling of image data [7, 2] we decided to evaluate the performance of our BoVW by applying it first to a LDA based topic model, and using the results in other domains of CV.

Next, we feed the visual code book as input to a LDA based topic model, along with quantized feature vector. The result is a probability distribution which gives the probability of occurrence of a particular visual word in the top 10 topics. The figure 14 shows the top 10 topics and the 10 most probable visual words that might occur in those topics (denoted by ϕ).

To evaluate the results, we have also found the ”burst probability” of a topic, that is the probability that the topic is composed of burst images. The figure 15 shows the topic burst probabilities, of the top 10 topics in the dataset.

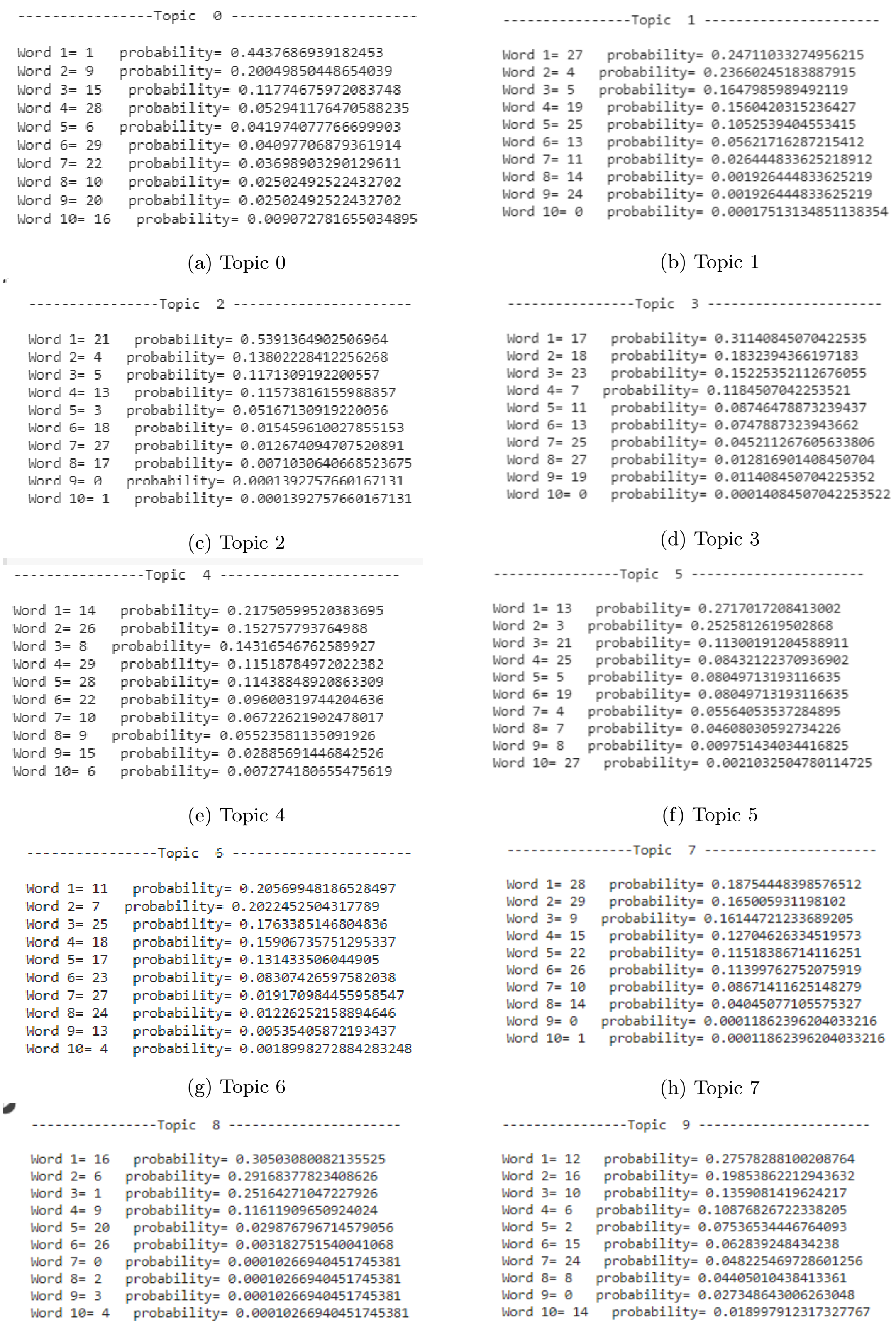
2.5 Training a classifier

Topic modelling on the feature descriptors,gives ϕ and θ. We have discussed the significance of ϕ in the previous sub section. Here, we shall focus on θ. Actually, θ gives us the image - topic distribution (for our case, it gives the segment - topic distribution), that is the probability with which a particular topic contributes to the generation of an image segment. It is a vector of length = 10 (for the top 10 topics), where each value is a probability that the topic contributes to a particular segment.

Next, in order to evaluate the effectiveness of the visual words, we have also trained a classifier to classify the images in between burst and non burst. We have tried classification both in the supervised way (using a naive approach and also using the topic modelling results) and unsupervised way.

2.5.1 Supervised Image Classification : Naive approach

From the earlier section of topic modelling results, we have already obtained the burst probability of each topic. Here, we will obtain the burst probability of each image segment, that is the probability that an image segment belongs to the class of burst images, or non burst images, as per the following calculation : Considering the segment number 10, from the list of all the segmented



```
----------------Topic  0 ----------------------

Word 1= 1   probability= 0.4437686939182453
Word 2= 9   probability= 0.20049850448654039
Word 3= 15   probability= 0.11774675972083748
Word 4= 28   probability= 0.052941176470588235
Word 5= 6   probability= 0.041974077766699903
Word 6= 29   probability= 0.04097706879361914
Word 7= 22   probability= 0.03698903290129611
Word 8= 10   probability= 0.02502492522432702
Word 9= 20   probability= 0.02502492522432702
Word 10= 16   probability= 0.009072781655034895
```

(a) Topic 0

```
----------------Topic  1 ----------------------

Word 1= 27   probability= 0.24711033274956215
Word 2= 4   probability= 0.23660245183887915
Word 3= 5   probability= 0.1647985989492119
Word 4= 19   probability= 0.1560420315236427
Word 5= 25   probability= 0.1052539404553415
Word 6= 13   probability= 0.05621716287215412
Word 7= 11   probability= 0.026444833625218912
Word 8= 14   probability= 0.001926444833625219
Word 9= 24   probability= 0.001926444833625219
Word 10= 0   probability= 0.00017513134851138354
```

(b) Topic 1

```
----------------Topic  2 ----------------------

Word 1= 21   probability= 0.5391364902506964
Word 2= 4   probability= 0.13802228412256268
Word 3= 5   probability= 0.1171309192200557
Word 4= 13   probability= 0.11573816155988857
Word 5= 3   probability= 0.05167130919220056
Word 6= 18   probability= 0.015459610027855153
Word 7= 27   probability= 0.012674094707520891
Word 8= 17   probability= 0.0071030640668523675
Word 9= 0   probability= 0.0001392757660167131
Word 10= 1   probability= 0.0001392757660167131
```

(c) Topic 2

```
----------------Topic  3 ----------------------

Word 1= 17   probability= 0.31140845070422535
Word 2= 18   probability= 0.1832394366197183
Word 3= 23   probability= 0.15225352112676055
Word 4= 7   probability= 0.1184507042253521
Word 5= 11   probability= 0.08746478873239437
Word 6= 13   probability= 0.0747887323943662
Word 7= 25   probability= 0.045211267605633806
Word 8= 27   probability= 0.012816901408450704
Word 9= 19   probability= 0.011408450704225352
Word 10= 0   probability= 0.00014084507042253522
```

(d) Topic 3

```
----------------Topic  4 ----------------------

Word 1= 14   probability= 0.21750599520383695
Word 2= 26   probability= 0.152757793764988
Word 3= 8   probability= 0.14316546762589927
Word 4= 29   probability= 0.11518784972022382
Word 5= 28   probability= 0.11438848920863309
Word 6= 22   probability= 0.09600319744204636
Word 7= 10   probability= 0.06722621902478017
Word 8= 9   probability= 0.05523581135091926
Word 9= 15   probability= 0.02885691446842526
Word 10= 6   probability= 0.007274180655475619
```

(e) Topic 4

```
----------------Topic  5 ----------------------

Word 1= 13   probability= 0.2717017208413002
Word 2= 3   probability= 0.2525812619502868
Word 3= 21   probability= 0.11300191204588911
Word 4= 25   probability= 0.08432122370936902
Word 5= 5   probability= 0.08049713193116635
Word 6= 19   probability= 0.08049713193116635
Word 7= 4   probability= 0.05564053537284895
Word 8= 7   probability= 0.04608030592734226
Word 9= 8   probability= 0.009751434034416825
Word 10= 27   probability= 0.0021032504780114725
```

(f) Topic 5

```
----------------Topic  6 ----------------------

Word 1= 11   probability= 0.20569948186528497
Word 2= 7   probability= 0.2022452504317789
Word 3= 25   probability= 0.1763385146804836
Word 4= 18   probability= 0.15906735751295337
Word 5= 17   probability= 0.131433506044905
Word 6= 23   probability= 0.08307426597582038
Word 7= 27   probability= 0.019170984455958547
Word 8= 24   probability= 0.01226252158894646
Word 9= 13   probability= 0.00535405872193437
Word 10= 4   probability= 0.0018998272884283248
```

(g) Topic 6

```
----------------Topic  7 ----------------------

Word 1= 28   probability= 0.18754448398576512
Word 2= 29   probability= 0.165005931198102
Word 3= 9   probability= 0.16144721233689205
Word 4= 15   probability= 0.12704626334519573
Word 5= 22   probability= 0.11518386714116251
Word 6= 26   probability= 0.11399762752075919
Word 7= 10   probability= 0.08671411625148279
Word 8= 14   probability= 0.04045077105575327
Word 9= 0   probability= 0.00011862396204033216
Word 10= 1   probability= 0.00011862396204033216
```

(h) Topic 7

```
----------------Topic  8 ----------------------

Word 1= 16   probability= 0.30503080082135525
Word 2= 6   probability= 0.29168377823408626
Word 3= 1   probability= 0.25164271047227926
Word 4= 9   probability= 0.11611909650924024
Word 5= 20   probability= 0.029876796714579056
Word 6= 26   probability= 0.003182751540041068
Word 7= 0   probability= 0.00010266940451745381
Word 8= 2   probability= 0.00010266940451745381
Word 9= 3   probability= 0.00010266940451745381
Word 10= 4   probability= 0.00010266940451745381
```

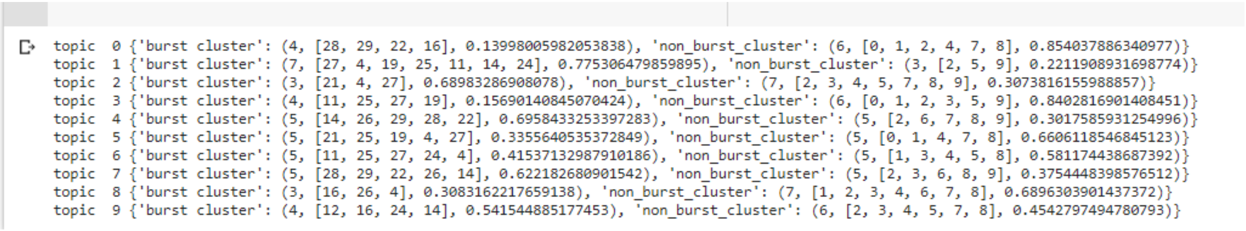
(i) Topic 8

```
----------------Topic  9 ----------------------

Word 1= 12   probability= 0.27578288100208764
Word 2= 16   probability= 0.19853862212943632
Word 3= 10   probability= 0.1359081419624217
Word 4= 6   probability= 0.10876826722338205
Word 5= 2   probability= 0.07536534446764093
Word 6= 15   probability= 0.062839248434238
Word 7= 24   probability= 0.048225469728601256
Word 8= 8   probability= 0.04405010438413361
Word 9= 0   probability= 0.027348643006263048
Word 10= 14   probability= 0.018997912317327767
```

(j) Topic 9

Figure 14: Top 10 topics generated using LDA based topic model along with the visual word - topic probability, ϕ



```
topic  0 {'burst cluster': (4, [28, 29, 22, 16], 0.13998005982053838), 'non_burst_cluster': (6, [0, 1, 2, 4, 7, 8], 0.854037886340977)}
topic  1 {'burst cluster': (7, [27, 4, 19, 25, 11, 14, 24], 0.775306479859895), 'non_burst_cluster': (3, [2, 5, 9], 0.2211908931698774)}
topic  2 {'burst cluster': (3, [21, 4, 27], 0.68983286908078), 'non_burst_cluster': (7, [2, 3, 4, 5, 7, 8, 9], 0.3073816155988857)}
topic  3 {'burst cluster': (4, [11, 25, 27, 19], 0.15690140845070424), 'non_burst_cluster': (6, [0, 1, 2, 3, 5, 9], 0.8402816901408451)}
topic  4 {'burst cluster': (5, [14, 26, 29, 28, 22], 0.6958433253397283), 'non_burst_cluster': (5, [2, 6, 7, 8, 9], 0.3017585931254996)}
topic  5 {'burst cluster': (5, [21, 25, 19, 4, 27], 0.3355640535372849), 'non_burst_cluster': (5, [0, 1, 4, 7, 8], 0.6606118546845123)}
topic  6 {'burst cluster': (5, [11, 25, 27, 24, 4], 0.41537132987910186), 'non_burst_cluster': (5, [1, 3, 4, 5, 8], 0.581174438687392)}
topic  7 {'burst cluster': (5, [28, 29, 22, 26, 14], 0.622182680901542), 'non_burst_cluster': (5, [2, 3, 6, 8, 9], 0.3754448398576512)}
topic  8 {'burst cluster': (3, [16, 26, 4], 0.3083162217659138), 'non_burst_cluster': (7, [1, 2, 3, 4, 6, 7, 8], 0.6896303901437372)}
topic  9 {'burst cluster': (4, [12, 16, 24, 14], 0.541544885177453), 'non_burst_cluster': (6, [2, 3, 4, 5, 7, 8], 0.4542797494780793)}
```

Figure 15: The dictionary sums up the results of topic modelling. The dictionary has 2 sub dictionaries, namely burst and non burst. Each of these sub dictionaries contains a tuple, having 3 things - the number of words in the topic i that belong to that class(burst or non burst), the common cluster indices which belong to the topic i and also to the burst/non burst clusters and the probability by which the topic represents burst or non burst images. The probabilities have been obtained by adding the individual word probabilities from the top 10 words.

images together,the probability that the segment belongs to the burst class of images is given by :

$$\Sigma(\theta[10][i]) \times [burstprobability]_{topic_i} \text{ , where i } \epsilon \text{ 0,1,2,3,4,5,6,7,8,9}$$

Then, a threshold value is chosen randomly, such that all the images that lie above the threshold value are classified as burst images and all images below the threshold value are classified as non burst images. We applied various threshold values within a range of [0,1] randomly and best results were obtained at a threshold of 0.65 - 0.7. The Figure 16 shows the performance of this naive classification approach for the given burst dataset, according to some standard metrics (Accuracy (a), Precision score(b), Recall score (c) and F1 score (d)). The confusion matrix corresponding to the results of each threshold value is also shown in figure 17.

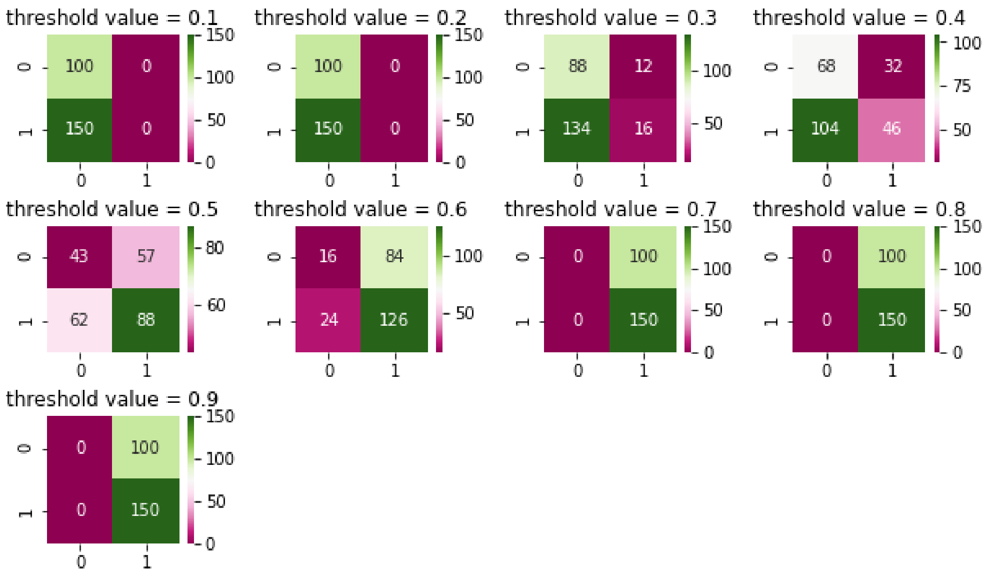
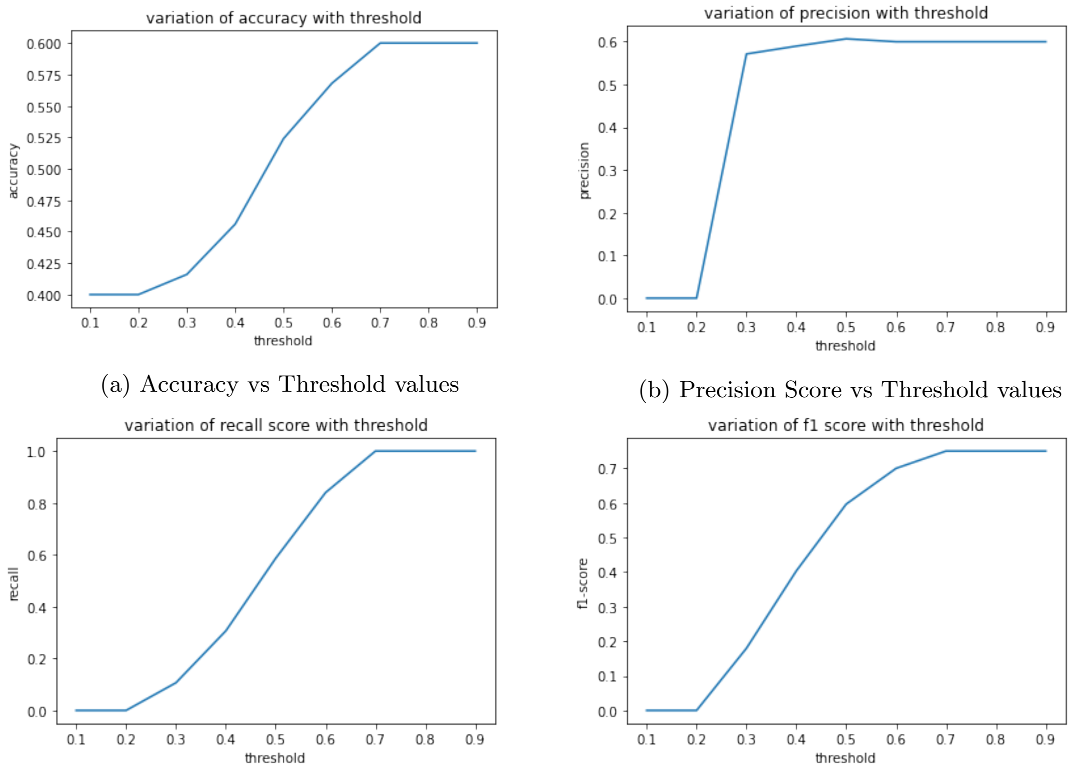


Figure 16: Confusion Matrix corresponding to different threshold values



(a) Accuracy vs Threshold values

(b) Precision Score vs Threshold values

(c) Recall Score vs Threshold values

(d) F1 score vs Threshold values

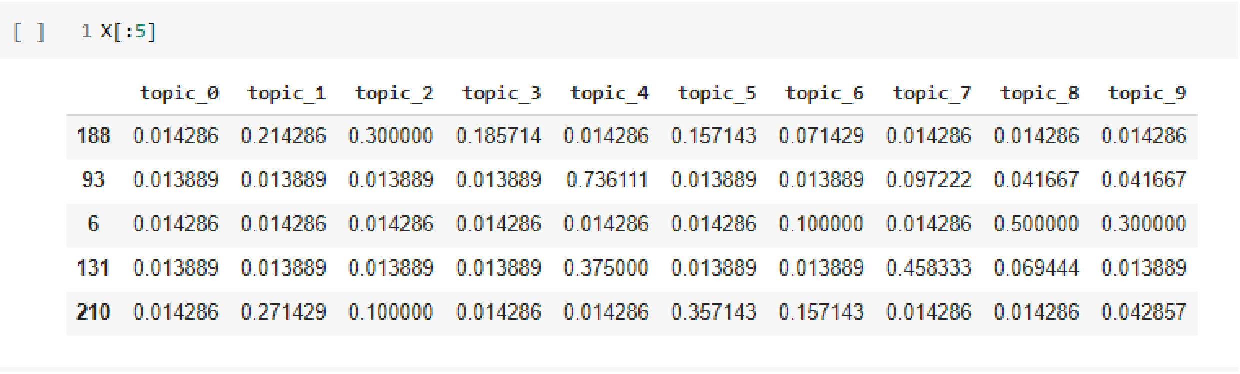
Figure 17: Performance of Naive classification approach based on standard metrics

2.5.2 Supervised Image Classification

We know, that topic modelling of data is capable of capturing the hidden semantic meaning between several data points. The objective of this approach was to check, if the captured semantic meaning from topic modelling results can yield better performance in supervised classification tasks.

In this approach, we are applying Naive Bayes classifier to classify images based on topic modelling results. For preparing the data, we have concatenated the θ, that is probability of an image segment belonging to the top 10 topics and formed a dataframe in Python, shown in figure 18. The output column is whether the image belongs to burst class of images (denoted by 1) or non burst class (denoted by 0).

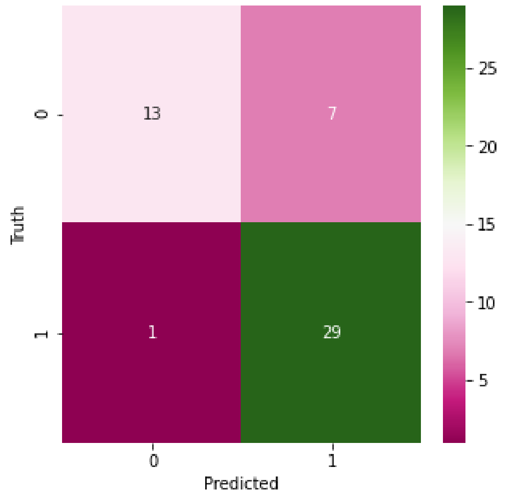
The dataset was separated into training and test sets in the ratio of 8 : 2, and multinomial Naive Bayes classifier was applied to classify the images. The performance of this technique as per some standard metrics is shown in figure 19. The confusion Matrix of the test dataset (50 records) is visualised in 19(a) and the accuracy, precision, recall and F1 score are displayed in fig 19(b)



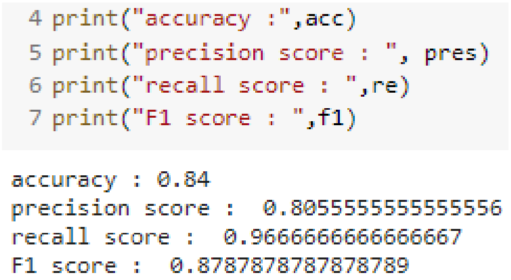
```
[ ]  X[:5]
```

	topic_0	topic_1	topic_2	topic_3	topic_4	topic_5	topic_6	topic_7	topic_8	topic_9
188	0.014286	0.214286	0.300000	0.185714	0.014286	0.157143	0.071429	0.014286	0.014286	0.014286
93	0.013889	0.013889	0.013889	0.013889	0.736111	0.013889	0.013889	0.097222	0.041667	0.041667
6	0.014286	0.014286	0.014286	0.014286	0.014286	0.014286	0.100000	0.014286	0.500000	0.300000
131	0.013889	0.013889	0.013889	0.013889	0.375000	0.013889	0.013889	0.458333	0.069444	0.013889
210	0.014286	0.271429	0.100000	0.014286	0.014286	0.357143	0.157143	0.014286	0.014286	0.042857

Figure 18: Dataset for classification using topic modelling results



(a) Confusion Matrix



```
print("accuracy :",acc)
print("precision score : ", pres)
print("recall score : ",re)
print("F1 score : ",f1)
```

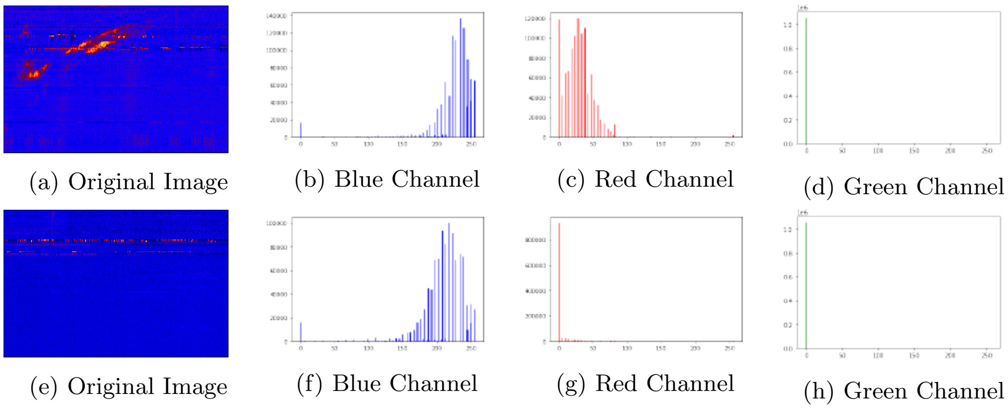
```
accuracy : 0.84
precision score :  0.8055555555555556
recall score :  0.9666666666666667
F1 score :  0.8787878787878789
```

(b) Other Metrics

Figure 19: Performance of classification based on topic modelling results

2.5.3 Unsupervised Image Classification

One very obvious way to distinguish the images (for our given dataset) is to exploit the difference in the pixel intensity values of the image. In case of non burst images, most of the images will have pixel intensities of blue color, thus, the histograms of the red and green color channels will show a single stick or very small peak. Whereas in case of burst images, since a significant portion of the image contains red - yellowish tone, the histogram of the red color channel for non burst images will show a significantly high peak and wider distribution of intensity values.Figure 20 shows the histogram of pixel intensities for a burst and non burst image.



(a) Original Image (b) Blue Channel (c) Red Channel (d) Green Channel

(e) Original Image (f) Blue Channel (g) Red Channel (h) Green Channel

Figure 20: Top : histogram of pixel intensities of a randomly picked burst image. Bottom : Histogram of pixel intensities of a randomly picked Non burst image

But, we will not use this technique, because first, it may not be applicable to other datasets, and secondly, our objective is to apply the visual words for image classification, such that the effectiveness and interpretability of these words can be evaluated.

Thus, we have used the topic modelling results to perform unsupervised clustering on the images as well. As we know, topic modelling on the image features gives us the following information:

- ϕ or the visual word - topic distribution, for the top 10, 20 or 30 topics. Here, we will use the top 10 topics.
- θ or the segmented image - topic distribution, for the 250 segments.

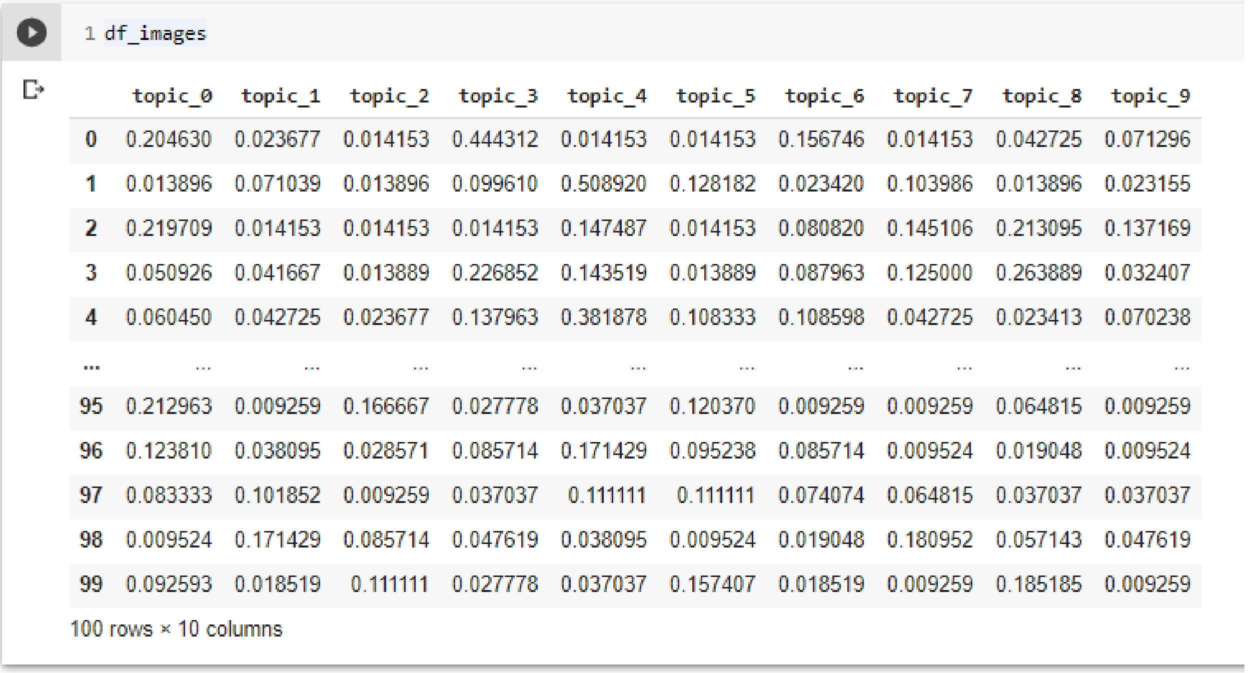
For the purpose of unsupervised classification, we have used θ. Say, the segments $S_k, S_{k+1}, S_{k+2}, \ldots$ belong to the image I_n. For each of these segments we have θ which gives us the possibility of the segment belonging to a particular topic T_i, for the top 10 topics. Then, the image - topic distribution is given as :

$$\theta_{I_n} = \Sigma\theta_{S_k} \text{ where k is the number of segments.}$$
$$Normalized\theta_{I_n} = \frac{\theta_{I_n}}{k}$$

This value was computed for all the images in the dataset and a data frame was created, which is shown in figure 21.

Then, K means clustering was applied on this dataframe, with number of clusters = 2 (for the 2 class labels, burst and non burst). 31 and 69 images were obtained for $Cluster_0$ and $Cluster_1$ respectively. But, we still did not know which cluster belonged to burst images and which one belonged to non burst images. Thus, we plotted these clusters, which is shown in figure 22 and 23.

From the plotted output, it was quiet evident that the $Cluster_0$ represented burst images and $Cluster_1$ represented non burst images, as most of the images in them were burst and non burst respectively.



```
df_images
```

	topic_0	topic_1	topic_2	topic_3	topic_4	topic_5	topic_6	topic_7	topic_8	topic_9
0	0.204630	0.023677	0.014153	0.444312	0.014153	0.014153	0.156746	0.014153	0.042725	0.071296
1	0.013896	0.071039	0.013896	0.099610	0.508920	0.128182	0.023420	0.103986	0.013896	0.023155
2	0.219709	0.014153	0.014153	0.014153	0.147487	0.014153	0.080820	0.145106	0.213095	0.137169
3	0.050926	0.041667	0.013889	0.226852	0.143519	0.013889	0.087963	0.125000	0.263889	0.032407
4	0.060450	0.042725	0.023677	0.137963	0.381878	0.108333	0.108598	0.042725	0.023413	0.070238
...	...	...	...	...	...	...	...	...	...	...
95	0.212963	0.009259	0.166667	0.027778	0.037037	0.120370	0.009259	0.009259	0.064815	0.009259
96	0.123810	0.038095	0.028571	0.085714	0.171429	0.095238	0.085714	0.009524	0.019048	0.009524
97	0.083333	0.101852	0.009259	0.037037	0.111111	0.111111	0.074074	0.064815	0.037037	0.037037
98	0.009524	0.171429	0.085714	0.047619	0.038095	0.009524	0.019048	0.180952	0.057143	0.047619
99	0.092593	0.018519	0.111111	0.027778	0.037037	0.157407	0.018519	0.009259	0.185185	0.009259

100 rows × 10 columns

Figure 21: Dataframe for image - topic distribution, created for the top 10 topics.

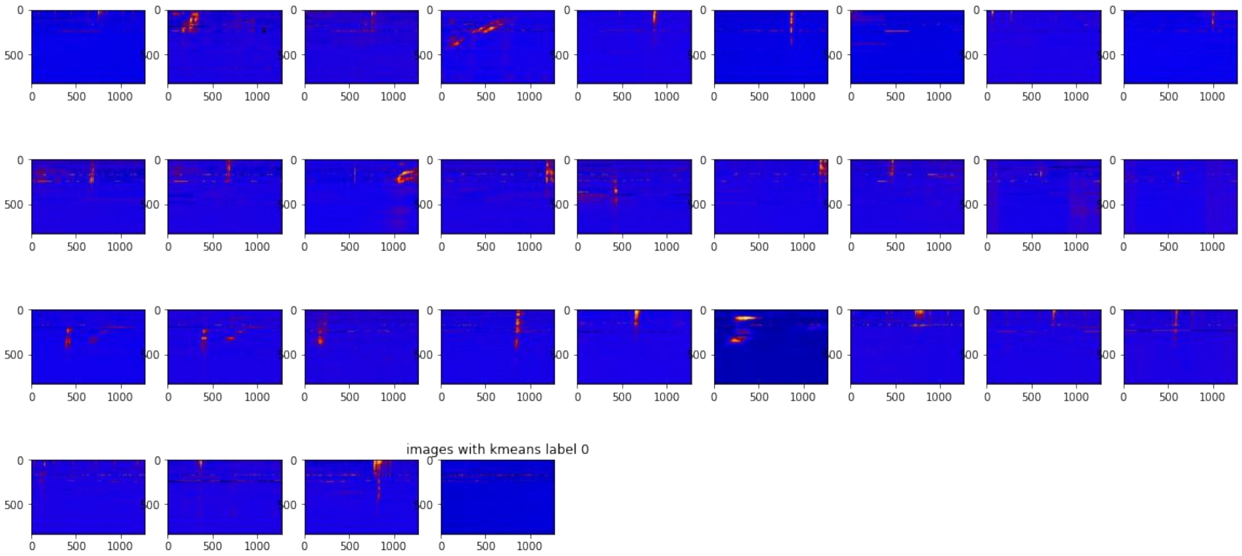


Figure 22: Images grouped in cluster 0. It can be seen that these are mostly burst images. Thus, we are considering '0' as the label for burst class of images.

Finally, we took this information as the "$Y_{prediction}$" and the original image labels (burst : 0, non burst : 1) as "Y_{test}", this information was used to test the performance of the unsupervised image classification, which is displayed in figure 24.

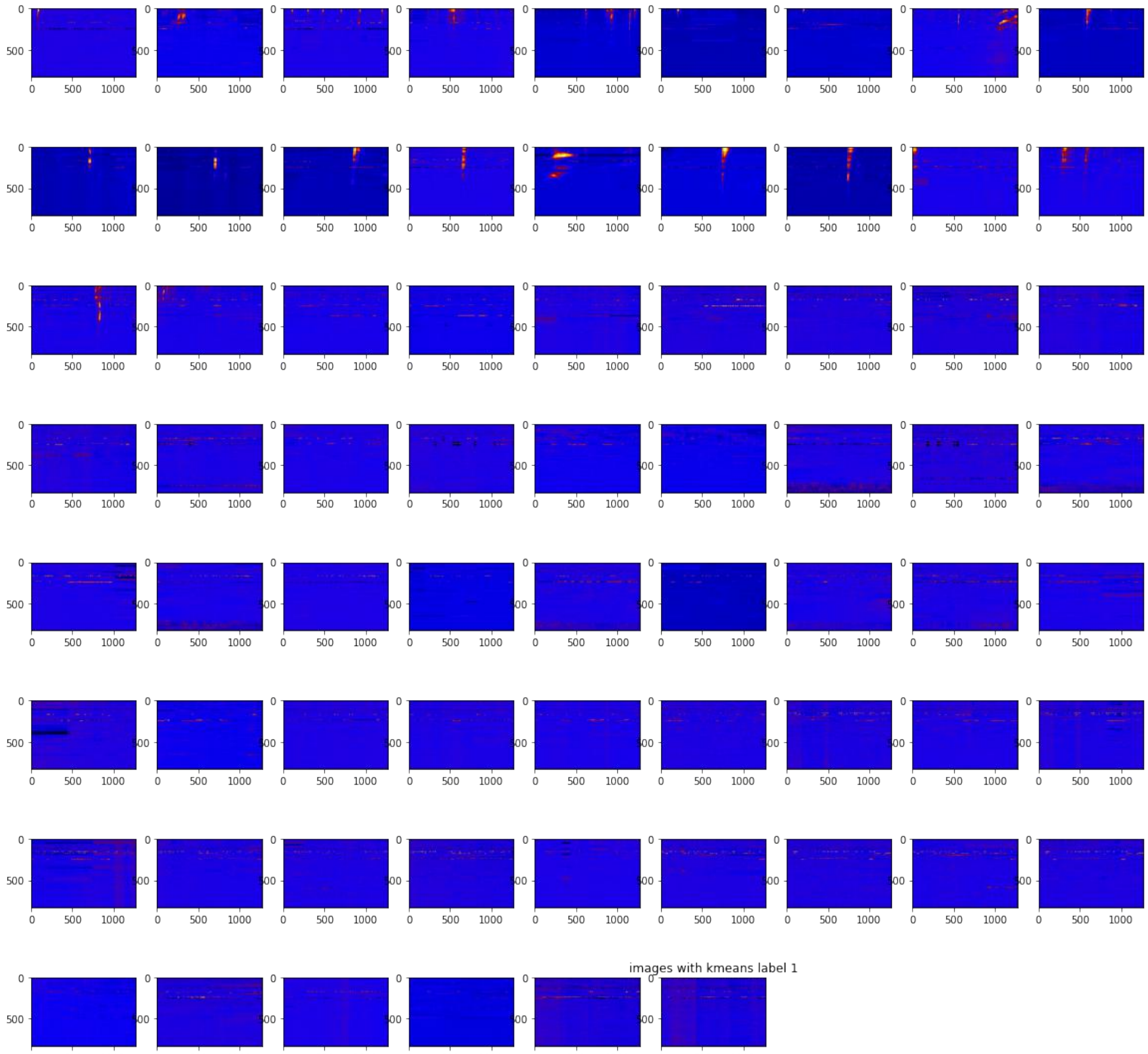
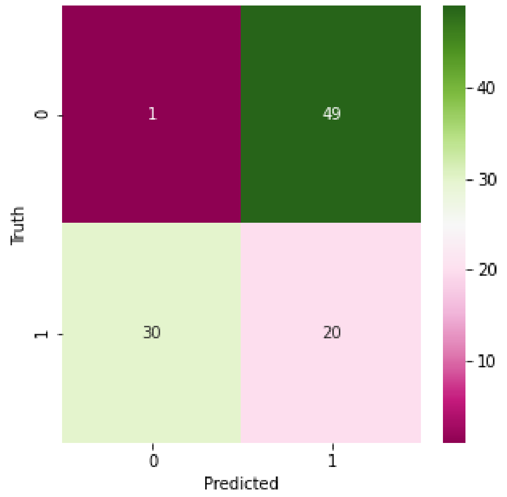
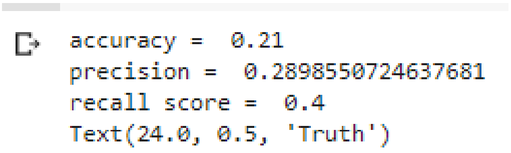


Figure 23: Images grouped in cluster 1. It can be seen that these are mostly non burst images. Thus, we are considering '1' as the label for non burst class of images.



(a) Confusion Matrix



```
accuracy =  0.21
precision =  0.2898550724637681
recall score =  0.4
Text(24.0, 0.5, 'Truth')
```

(b) Other Metrics

Figure 24: Performance of the unsupervised image classification

3 *Conclusion and Future Scope*

3.1 Conclusion

The objective of this project was to improve the basic BoVW algorithm, in order to generate ”effective visual words” from image data. We were successful in our task, which was quiet evident in the classification results. Classification of the images using a naive thresholding approach gave us 52-55 % classification accuracy, whereas topic modelling results gave us a much higher accuracy of 84%. We have also performed unsupervised classification, but results indicate need for improvement.

- For image Segmentation, we tested several clustering algorithms like K means clustering, Gaussian Mixture Models, Simple thresholding, watershed algorithm and contour detection, etc.
- We studied several feature extraction algorithms like SIFT, ORB and FAST.
- We worked on how the topic modelling results can be applied in CV domains, and learnt that the semantic gap that occurs in the BoVW can be overcome using the topic modelling results.

However, there is still a huge scope of improvement in the aforesaid work.

3.2 Future Scope

There are still several approaches/modifications that we wish to try, but due to the lack of time, these are left for future exploration. They are as follows:

- Firstly, as observed, the classification results (especially in an unsupervised setting) are not near perfect. That is, there are too many false negatives (too many burst images identified as non burst). This shows that the semantic gap between Mid level and High level features, was narrowed down, but not totally diminished.
- Secondly, while forming the visual codebook, we have used k means clustering to quantize the feature space, which gave the best results when 31 clusters were used, however weighted k means (which uses soft assignments) converged at n = 10. This could be used for better performance and lesser computational load.
- Some more feature extraction algorithms could be tried, like BRIEF [?] and SURF [3].
- SIFT, though is scale and viewpoint invariant, is flip variant. That is, the SIFT features extracted from two images, which are exactly mirror reflections of each other are different. Flip invariant approaches to SIFT are discussed in [18, 13, ?] and can be applied for generating the BoVW. This could greatly improve the results, especially of unsupervised learning.
- Panoptic Image segmentation techniques should be attempted, as they give us the best of 2 worlds, that is, instance segmentation and semantic segmentation.

References

[1] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine Süsstrunk. Slic superpixels compared to state-of-the-art superpixel methods. *IEEE transactions on pattern analysis and machine intelligence*, 34(11):2274–2282, 2012.

[2] Kobus Barnard, Pinar Duygulu, David Forsyth, Nando De Freitas, David M Blei, and Michael I Jordan. Matching words and pictures. 2003.

[3] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. In *European conference on computer vision*, pages 404–417. Springer, 2006.

[4] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *the Journal of machine Learning research*, 3:993–1022, 2003.

[5] Lars Buitinck, Gilles Louppe, Mathieu Blondel, Fabian Pedregosa, Andreas Mueller, Olivier Grisel, Vlad Niculae, Peter Prettenhofer, Alexandre Gramfort, Jaques Grobler, Robert Layton, Jake VanderPlas, Arnaud Joly, Brian Holt, and Gaël Varoquaux. API design for machine learning software: experiences from the scikit-learn project. In *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, pages 108–122, 2013.

[6] Dawood Al Chanti and Alice Caplier. Improving bag-of-visual-words towards effective facial expressive image classification. *arXiv preprint arXiv:1810.00360*, 2018.

[7] Tao Chen, Hany SalahEldeen, Xiangnan He, Min-Yen Kan, and Dongyuan Lu. Velda: Relating an image tweet's text and images. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 29, 2015.

[8] Pedro Costa and Aurélio Campilho. Convolutional bag of words for diabetic retinopathy detection from eye fundus images. *IPSJ Transactions on Computer Vision and Applications*, 9(1):1–6, 2017.

[9] Mouna Dammak, Mahmoud Mejdoub, and Chokri Ben Amar. A survey of extended methods to the bag of visual words for image categorization and retrieval. In *2014 International Conference on Computer Vision Theory and Applications (VISAPP)*, volume 2, pages 676–683. IEEE, 2014.

[10] Ivica Dimitrovski, Dragi Kocev, Suzana Loskovska, and Sašo Džeroski. Improving bag-of-visual-words image retrieval with predictive clustering trees. *Information Sciences*, 329:851–865, 2016.

[11] Spyros Gidaris, Andrei Bursuc, Nikos Komodakis, Patrick Pérez, and Matthieu Cord. Learning representations by predicting bags of visual words. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6928–6938, 2020.

[12] Spyros Gidaris, Andrei Bursuc, Gilles Puy, Nikos Komodakis, Matthieu Cord, and Patrick Pérez. Online bag-of-visual-words generation for unsupervised representation learning. *arXiv preprint arXiv:2012.11552*, 2020.

[13] Xiaojie Guo and Xiaochun Cao. Find: A neat flip invariant descriptor. In *2010 20th International Conference on Pattern Recognition*, pages 515–518. IEEE, 2010.

[14] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

[15] Alexander Kirillov, Kaiming He, Ross Girshick, Carsten Rother, and Piotr Dollár. Panoptic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9404–9413, 2019.

[16] Xiaobin Liu, Shiliang Zhang, Tiejun Huang, and Qi Tian. E2bows: An end-to-end bag-of-words model via deep convolutional neural network for image retrieval. *Neurocomputing*, 395:188–198, 2020.

[17] G Lowe. Sift-the scale invariant feature transform. *Int. J*, 2(91-110):2, 2004.

[18] Rui Ma, Jian Chen, and Zhong Su. Mi-sift: Mirror and inversion invariant generalization for sift descriptor. In *Proceedings of the ACM International Conference on Image and Video Retrieval*, pages 228–235, 2010.

[19] Edward Rosten and Tom Drummond. Machine learning for high-speed corner detection. In *European conference on computer vision*, pages 430–443. Springer, 2006.

[20] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. Orb: An efficient alternative to sift or surf. In *2011 International conference on computer vision*, pages 2564–2571. Ieee, 2011.

[21] Qi Wang, Jungang Xu, Hong Chen, and Ben He. Two improved continuous bag-of-word models. In *2017 International Joint Conference on Neural Networks (IJCNN)*, pages 2851–2856. IEEE, 2017.

[22] Shiliang Zhang, Qi Tian, Gang Hua, Qingming Huang, and Wen Gao. Generating descriptive visual words and visual phrases for large-scale image applications. *IEEE Transactions on Image Processing*, 20(9):2664–2677, 2011.