Novel Classifier for Smart Traffic Network

Submitted June 2020, in partial fulfillment of
the conditions for the award of the degree **Bachelor of Computer Engineering/Computer Science.**

Thomas Courtney

3175353

Supervised by Assoc. Prof. Kaushik Mahata

School of Engineering

University of Newcastle

I hereby declare that this dissertation is all my own work, except as indicated in the text:

Signature _____

Date _____ / _____ / _____

Contents

List of Tables

List of Figures

Chapter 1

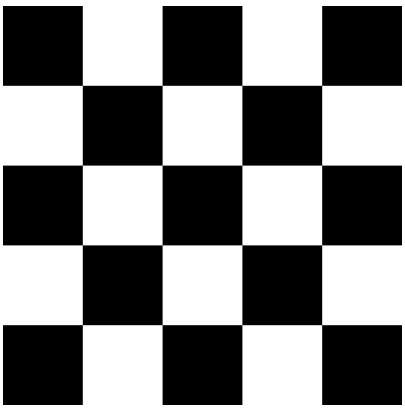
Literature Review

This section of the report expands upon the theory behind the design of the system. Digital image processing and computer vision are broad fields and so only concepts relevant to the system are explored.

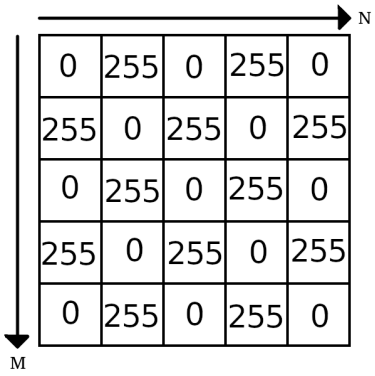
1.1 Digital Images

Digital images consists of a discrete number of discrete valued data points called pixels. It is the value of these pixels that determine their color. A pixel's value is also called its *intensity* and will lie within a range of values determined by the image's encoding. For example an 8-bit image has pixel values in the range $[0, 255]$.

Digital images are often represented as a 2D array of values where each cell corresponds to an intesnity. In software in particular, digital images are stored and manipulated as arrays. An image's array is $M \times N$ pixels in size, which is also its resolution. Notice that in Figure 1.1 the lowest value is 0 (black) and the highest value is 255 (white). All other colors that can be represented by 8-bits are between these two values.



(a)



(b)

Figure 1.1: (a) A Simple 8-bit Image (b) Array Representation of Image.

2D array representation allows pixels to be referenced by their relative position in the M and N directions. Furthermore, M and N may be substituted for the x and y axes of the Cartesian plane. In fact, a digital image is just a two dimensional function where each pixel is described by a coordinate (x,y) and an intensity (z) at that point.

$$z = f(x, y) \tag{1.1}$$

This means that two dimnesional operations may be applied to a digital image and images may be modelled as 3D surfaces. Modelling images as a surface is useful when trying to visualize rates of change in an image. In Figure 1.1 you can see that there is a rapid change in value between different coloured squares, shown as steep cliff edges.

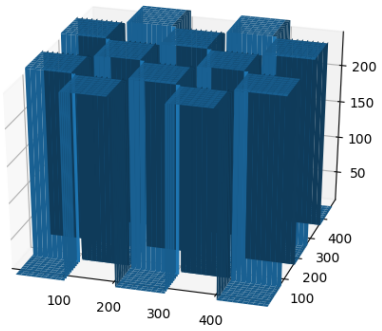


Figure 1.2: Surface Plot of Figure 1.1a

1.2 Linear Filtering

Filtering an image is a way to augment or extract information from it. Linear filters are a subset of operations that operate on a neighbourhood of pixels in an image. All linear filters operate on the same prinicpal, that is, they output a weighted average of their input. Filters take pixels in the neighbourhood covered by the filter as input. It is the weightings of a filter, known as filter coefficients, that determine the effect of the filter. These weights are stored in a matrix called a mask or kernel (see Figure 1.3) that's then convolved or correlated with an image (see sections 1.2.1 & 1.2.2).

$$Kernel = \frac{1}{\sum_{i=0}^{M}\sum_{j=0}^{N} c_{i,j}} \begin{bmatrix} c_{0,0} & c_{0,1} & \dots & c_{0,n} \\ c_{1,0} & c_{1,1} & \dots & c_{1,n} \\ \vdots & \vdots & \ddots & \vdots \\ c_{m,0} & c_{m,1} & \dots & c_{m,n} \end{bmatrix}$$

Figure 1.3: General Form of a Linear Filter

A filter kernel is nearly always square so as to have a center cell which sits atop a reference pixel. The result of the filter's application at that reference pixel will be stored in the output image at the location of the reference pixel. Notice in Figure 1.4 how the mask sits over the reference pixel.

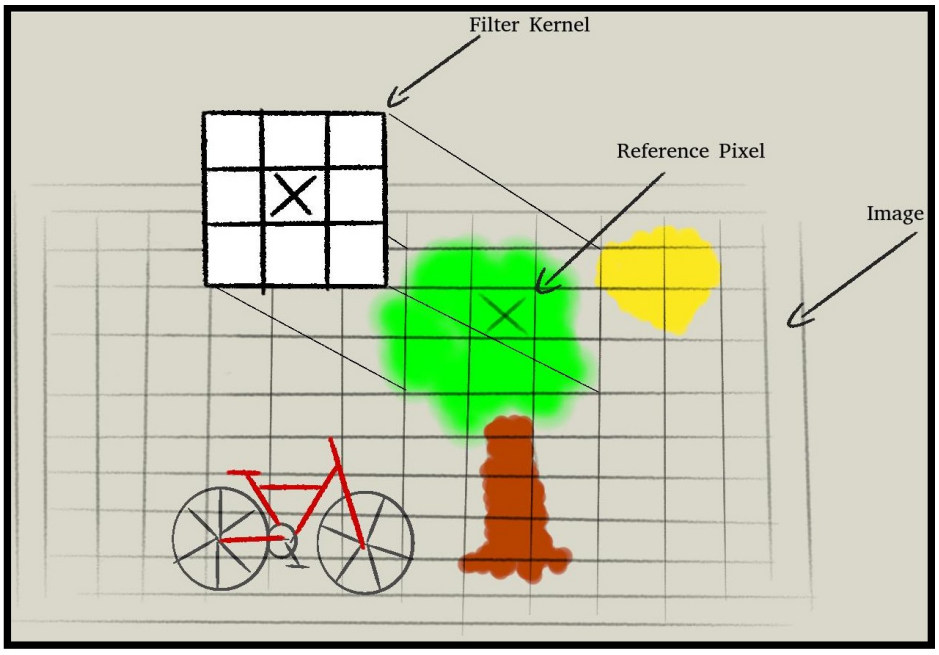
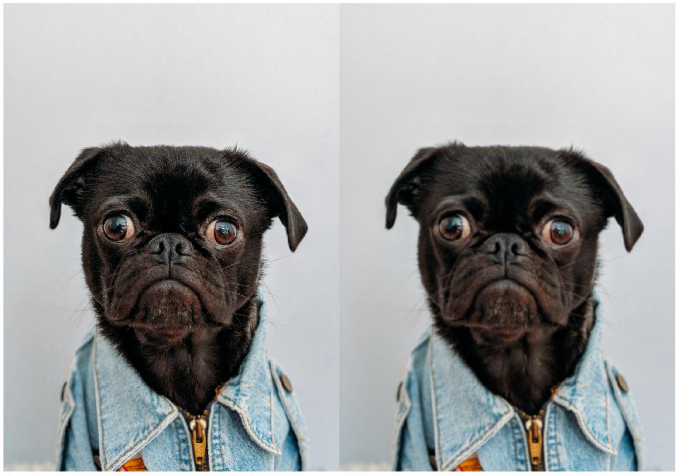


Figure 1.4: Visualization of a Filter Kernel Application

A 'box filter', for example, outputs the average of its inputs because the filter weights are evenly distributed. By passing this filter over an image its sharpness is reduced giving a smoothing or blurring effect. This can be observed in Figure 1.5a.



(a) Left: Photo by Charles Deluvio. Right: Application of 16×16 box filter.

$$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

(b) Box Filter Kernel.

Figure 1.5: Box Filter application and kernel.

1.2.1 Correlation

The application of a linear filter $h(u, v)$ to an image $f(i, j)$ may be described as follows

$$g(i,j) = \sum_{u=-k}^{k} \sum_{v=-l}^{l} f(i+u, j+v)h(u,v) \tag{1.2}$$

$g(i, j)$ is the output image. Performing correlation with a filter may be notated more concisely by the correlation operator.

$$g = f \otimes h$$

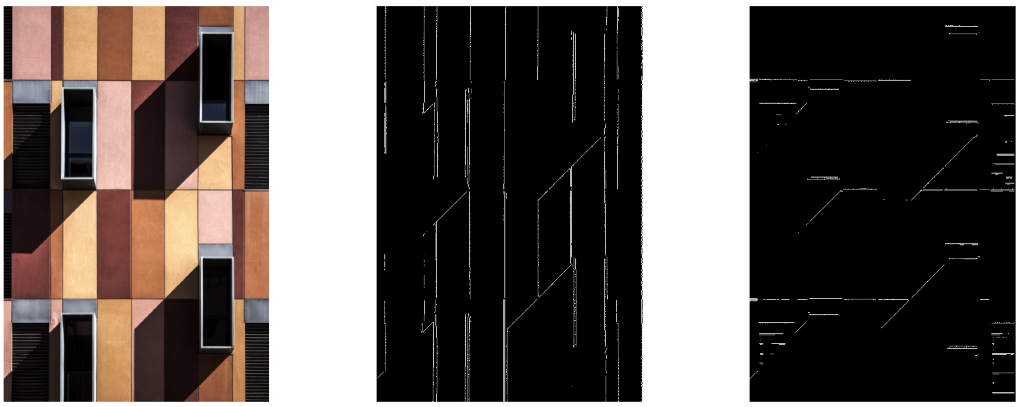
Correlation measures the similarity between two signals. Both digital images and linear filters are two dimensional signals. Performing correlation between them will yield an output image where the highest values correspond to where the image and filter were most similar [1]. This useful because if you wish to emphasise a feature in an image it can be done by correlating it with a filter that describes that feature. For example, if you wished to exagerate vertical and horizontal lines you could use Sobel filters as in 1.6.

$$\begin{bmatrix} -1 & -1 & -1 \\ 2 & 2 & 2 \\ -1 & -1 & -1 \end{bmatrix} \qquad \begin{bmatrix} -1 & 2 & -1 \\ -1 & 2 & -1 \\ -1 & 2 & -1 \end{bmatrix}$$

(a) Horizontal Sobel Filter Mask (b) Vertical Sobel Filter Mask

Figure 1.6: Sobel Filters

The use of negative weightings means that values next to an edge are diminished and the positively weighted line sections (the 2s) strengthen line features. Notice in figure 1.7 how the lines have a high values (white) and regions that aren't lines are low valued (black).



(a) Image by Simone Hutsch (b) Vertical Sobel Filter (c) Horizontal Sobel Filter

Figure 1.7: Application of Sobel filters to exagerate lines.

Correlation is *shift invariant*, which means that it does the same thing no matter where in an image it is applied. To satisfy this property correlation may be superpositioned

$$a(f_1 + f_2) = af_1 + af_2$$

and abides by the shift invariance principle

$$g(i,j) = f(i+k, j+l) \Leftrightarrow \ (h \circ g)(i,j) = (h \circ f)(i+k, j+l)$$

Correlation has the side effect of flipping both horizontally and vertically the location of output points relative to location the center point (*reference point*) in the original image which may be undesirable.

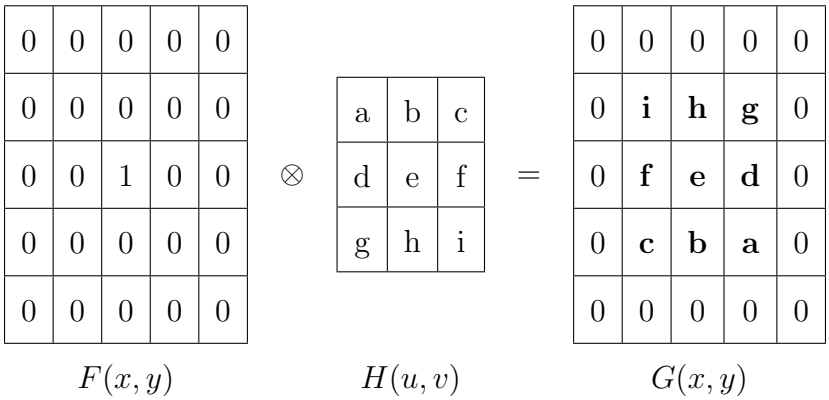


Figure 1.8: Correlation of a filter and an image.

1.2.2 Convolution

Convolution is also a linear operation that is shift invariant. It is very similar to correlation except that where correlation measure similarity between signals convolution measures the effect of one signal on another. It is described mathematically by the expression,

$$g(i,j) = \sum_{u=-k}^{k} \sum_{v=-l}^{l} f(u,v)h(i-u,j-v)$$

Notice that the filter $h(i,j)$ is rotated 180 degrees. This causes the output's orientation to match the original image. Convolution may be notated as follows,

$$g = f * h$$

0	0	0	0	0
0	0	0	0	0
0	0	1	0	0
0	0	0	0	0
0	0	0	0	0

$F(x,y)$

$*$

a	b	c
d	e	f
g	h	i

$H(u,v)$

$=$

0	0	0	0	0
0	**a**	**b**	**c**	0
0	**d**	**e**	**f**	0
0	**g**	**h**	**i**	0
0	0	0	0	0

$G(x,y)$

Figure 1.9: Convolution of a filter and an image.

1.3 The Gaussian

The Gaussian function is characterised by its symmetric bell shaped curve (Figure 1.10 & 1.11). It is defined by three parameters, its mean μ, standard deviation σ and amplitude A.

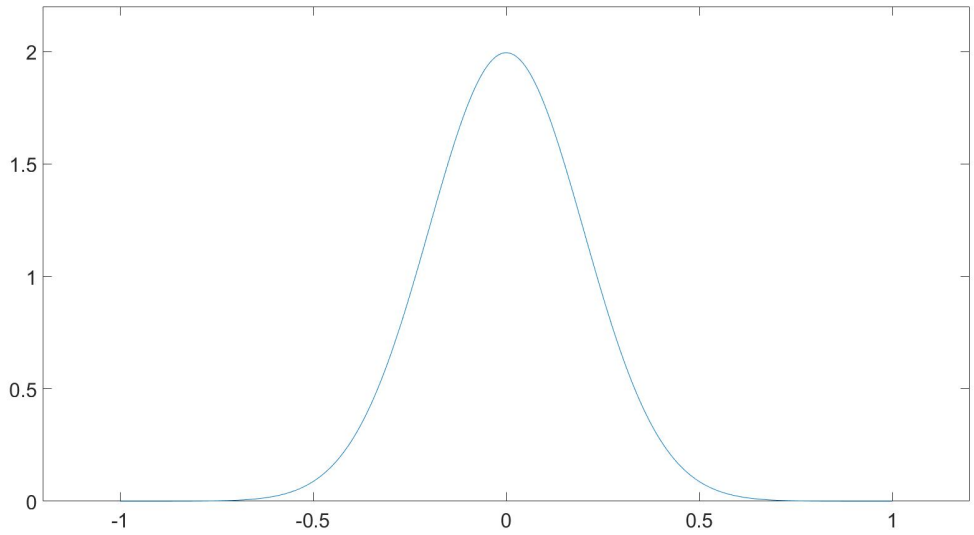


Figure 1.10: The Gaussian Function in 2D with μ=0, σ=0.2 and $A = 2$.

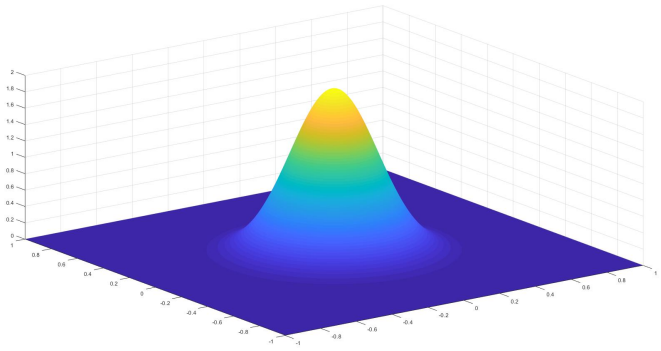


Figure 1.11: The Gaussian Function in 1D with μ=0, σ=0.2 and $A = 2$.

The Gaussian provides a consistent model for normal distributions, aka the bell curve. Normal distributions follow the *central limit theorem* where in if a histogram is taken of a sufficiently large number of independent random variable they will distribute with a central most probable value and symmetrically fall away either side of this value. Many datasets follow this trend closely enough that the Gaussian can be used to approximate a probability distribution for them.

The general one dimensional Gaussian is described:

$$f(x) = \frac{A}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}(\frac{x-\mu}{\sigma})^2} \tag{1.3}$$

The mean value, the central limit, of the sample distribution is μ. σ is the standard deviation or spread of the distribution, the greater the standard deviation the fatter the distribution. The amplitude A is the probability for a given value in the domain. In Figure 1.10, for example, the proability of a value being 0 is 2. Generally, the curve is used to determine a sample's probability desnity within a value range. This is calculated as the area under the curve using integration, generally expressed:

$$P(x) = \int_a^b \frac{A}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}(\frac{x-\mu}{\sigma})^2} dx \tag{1.4}$$

In reality data sets can have many dimensions and distributions need to be able to express all of them. The notation for a multi-variate Gaussian distribution, for a k-dimensional random vector $\boldsymbol{X} = (X_1, \ldots, X_k)^T$, is

$$\boldsymbol{X} \sim \mathcal{N}_k(\mu, \sigma^2) \tag{1.5}$$

1.4 Clustering

Clustering is a method of segmenting an image into disjoint sets known as classes. This is useful for separating types of features or objects in an image. For example in Figure 1.12 the toy car has been segmented from the background.

(a) Image by Gustavo, Upsplash.

(b) Segmentation of car from background.

Figure 1.12: Segmentation of toy cars using K-Means Clustering.

In an image every pixel can be individually classified or groups of pixels known as super-pixels may be treated as single entities. Entities are sorted based on their similarity. For example, a pixel's intensity is a feature that could be used to cluster it with other pixels of similar intensity. As seen in (1.6) an entity's comparable traits may be represented by a feature vector. An entity may have any number of features which as also regarded as dimensions, d.

$$\vec{v} = \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_n \end{bmatrix} \tag{1.6}$$

1.4.1 K-Means

There are many methods by which to cluster, K-Means is a popular method due to its speed and simplicity. Data points are sorted into one of K clusters where the average value of the cluster is stored in a central data point known as the cluster centroid. Centroids may initially be randomly generated or selected from the given data set.

Data points or samples are placed in the class of the centroid that is closest to them. The

distance between them is the Euclidean Distance between their feature vectors as in 1.7.

$$D(\vec{p}, \vec{q}) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \ldots + (p_n - q_n)^2} \tag{1.7}$$

K-Mean's algorithm proceeds as follows:

1. Place the initial K centroids.

2. Assign all data points to their nearest centroid.

3. Update the centroids' values to be the mean of all data points in their cluster.

4. Repeat steps 2 and 3 until a criteria is met, for example until cluster centroids no longer move (converge).

K-means benefits greatly from its low computational cost such that it is oten performed a number of times with different initial centroids and the instance best result is used. The best result is defined as having the smallest intracluster variance. K-Means is disadvantaged by the implicit trait that it formulates clusters of similar sizes. This happens because the algorithm seeks to minimize variance (spread) in each cluster hence the 'ideal' centroid placement will form distributions spherically about centroids. This method cannot disentangle overlapping samples that belong in different classes. In Figure 1.13 clear edges between clusters and sphereical distributions can be observed. The algorithm can only implement *hard assignments*, as opposed to a *soft assignment* that consider the probability of a sample's class membership.

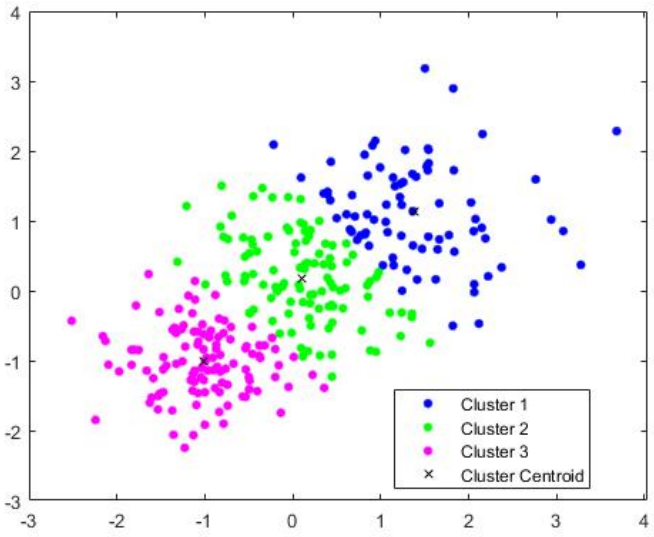
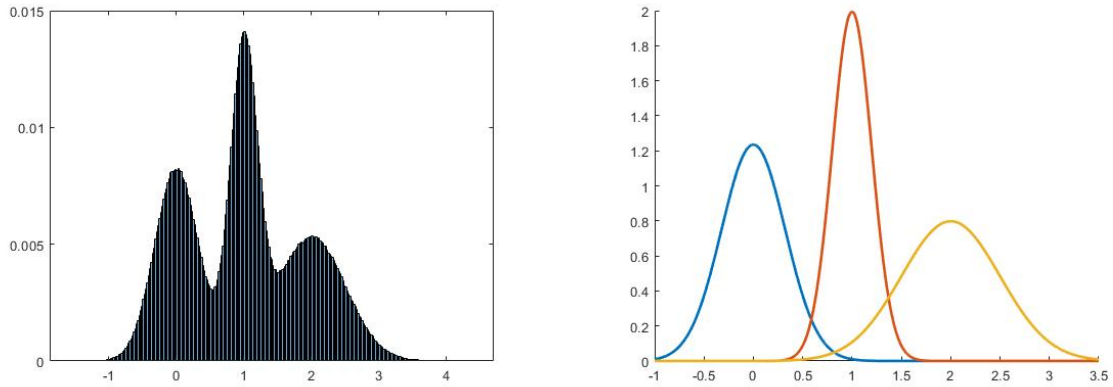


Figure 1.13: K-Means clustering performed on random data.

1.4.2 Mixtures of Gaussians

Mixtures of Gaussian or the Gaussian Mixture Model (GMM) is a method of clustering data that's able to disentangle ambiguous samples by considering a sample's probability of belonging to a class, known as a soft assignment.

Initially, K Gaussian probability density functions are randomly generated corresponding to K clusters. The Gaussians model the probability of a sample belonging to a corresponding cluster. By superpositioning all of the Gaussians a sample's complete probabilistic model is created, i.e. a model for *all* clusters. If a dataset has low dimensionality, by taking a histogram of its values the Gasussian's initial conditions can be approximated, as in Figure 1.14.



(a) Normalized Histogram of 1D samples. (b) Gaussians derived from 1D histogram.

Figure 1.14: Formulation of Mixture of Gaussians.

With each iteration of the GMM algorithm the parameters of the model's component Gaussians are tuned according to the covariance between samples in each cluster. The algorithm seeks the highest covariance possible in each of its clusters. It is by considering the covariance of samples that the GMM is able to best classify ambiguous samples. The higher the correlation between sample dimensions the more likely it is they belong in the same cluster. This can observed in Figure 1.15, the data being clustered is the same as in Figure 1.13 but notice the elliptical shape of the cluster distributions as opposed to the spherical shape of K-Means clustering.

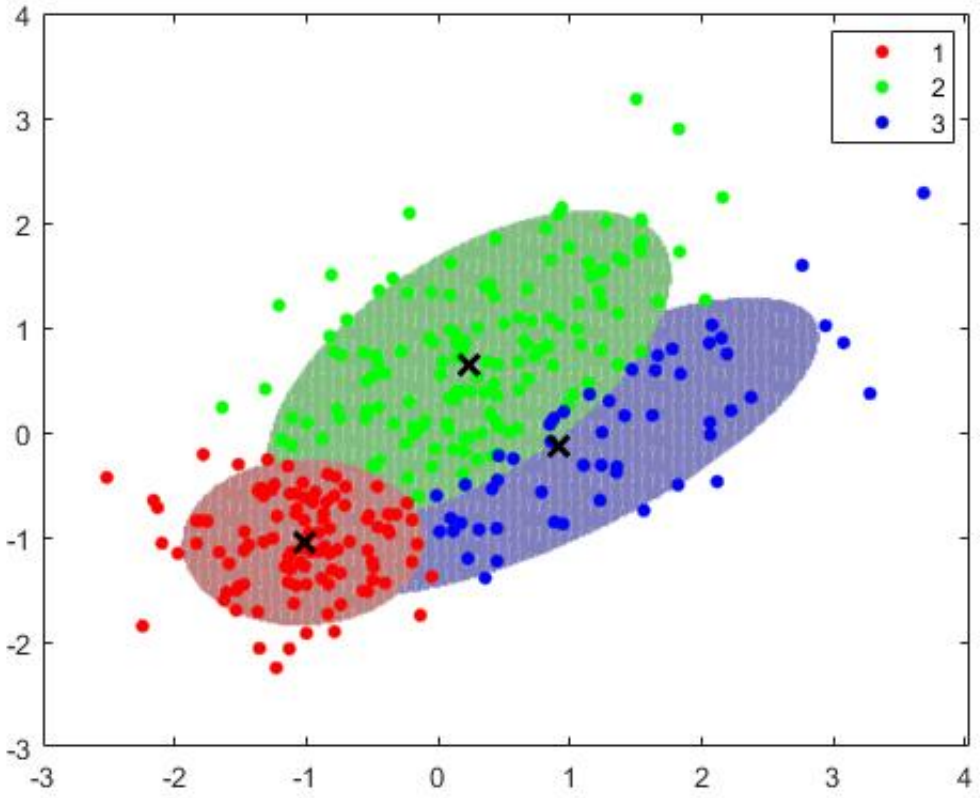


Figure 1.15: Clustering using GMM.

first by use of the Halanobis Distance or second by a changing of axes using eigenvalues according to covariance such that we can now just think of distances as Euclidean as in the K-Means met

With each iteration of the algorithm the three defining properties of the Gaussian components are updated, the mean,

Each cluster is modelled by a Gaussian that gives the probability density of a samples that belong to that class. The Gaussian is initially determined by taking a histogram of all values. For one dimensional data this would look something like Figure 1.14a. A histogram can be created for any N-dimensional data set but it becomes increasingly problematic to visualize with each additional dimension.

Figure 1.14b shows that the number of peaks in the histogram correspond to the number

of Gaussians it can be broken in to. Furthermore, each Gaussian represents a potential cluster in the dataset. In Figure 1.16 it can be seen that the local maxima have a domain value equal to the mean of the corresponding Gaussian, μ_k. By knowing the mean of the

The 'mixture' of Gaussians is the superposition of each of the individual Gaussians forming a new probability density function. Because a sample *must* belong to one of the classes the total area under the mixed model must be one.

In Figure 1.16 the arrows indicate the approximate location of the mean and standard deviation.

The values atop each peak are the weigthings for each Gaussian.

Notice the peaks in Figure 1.16 have been scaled such that the sum of weigthings is equal to one. This is because the probability of a sample belonging to at least one of the clusters must be equal to one. This is expressed in 1.8 and 1.9

$$0 \leq \pi_k \leq 1 \tag{1.8}$$

$$\sum_{k=1}^{K} \pi_k = 1 \tag{1.9}$$

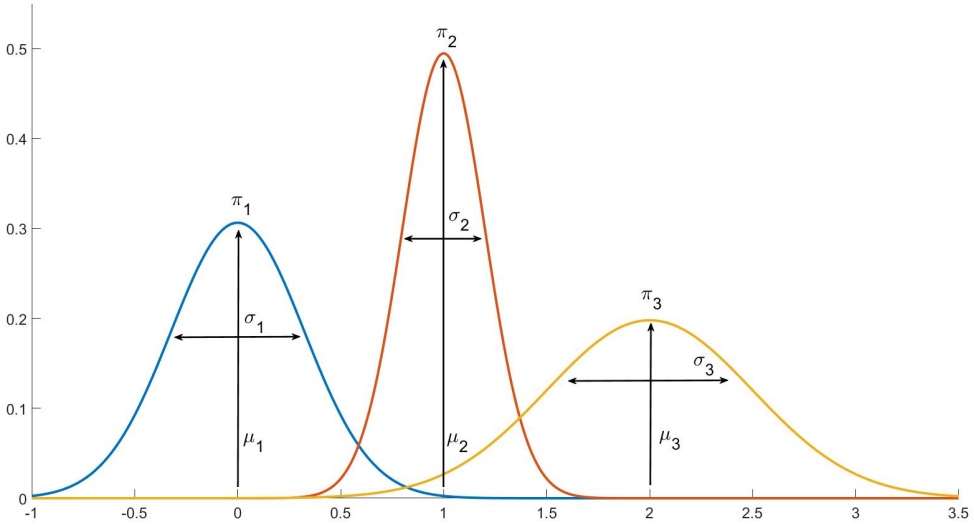


Figure 1.16: Individual Gaussians with scaled weightings.

This method is perhaps explained with an example.

Expectation Maximization

1.4.3 Background Subtraction

This is performed by identifying a background image and then subtracting this image from subsequent frames of video to determine what has changed and hence what is a foreground image.

1.4.4 Expectation Maximization

This is an optimization scheme used to fit a Gaussian Mixture Model. It is an iterative method that faurantees convergence to a local maximum in a search space.

Chapter 2

Design

The design of the system is compromised of a number of submodules.

[Insert Block Diagram of Submodules Comprising Whole System]

[Then Breakdown Each Submodule]

[Why was a CNN not used?]

Bibliography

[1] Sabyasachi Sahoo. Deciding optimal kernel size for cnn.