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 Kaushik Mahata ddd

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

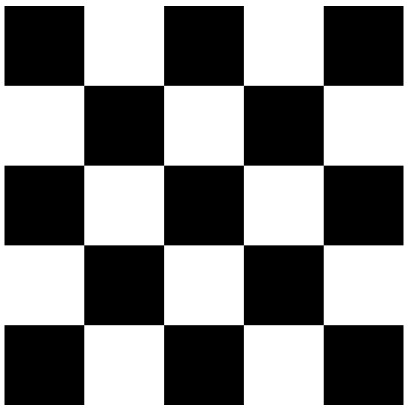
Literature Review

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.

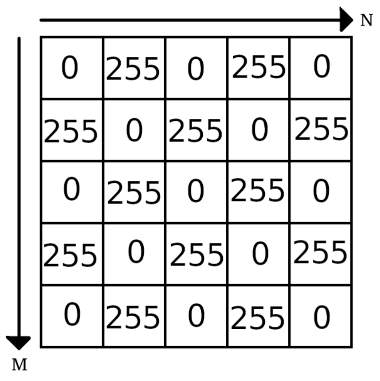
1.1 Digital Images

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)



0	255	0	255	0
255	0	255	0	255
0	255	0	255	0
255	0	255	0	255
0	255	0	255	0

(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.

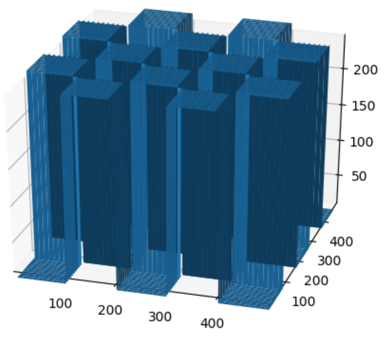


Figure 1.2: Surface Plot of Figure 1.1a

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).

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

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.

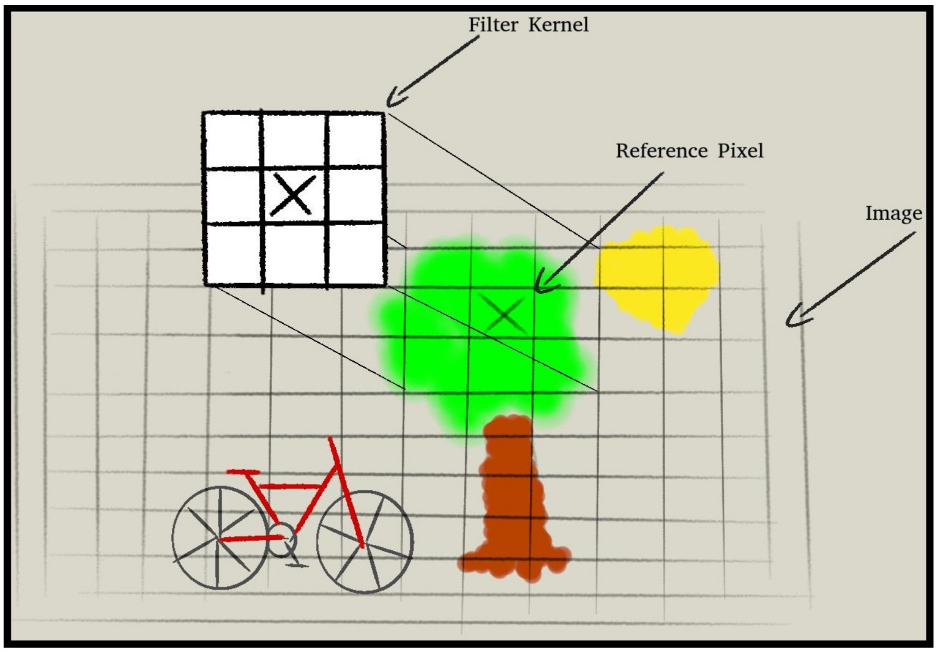
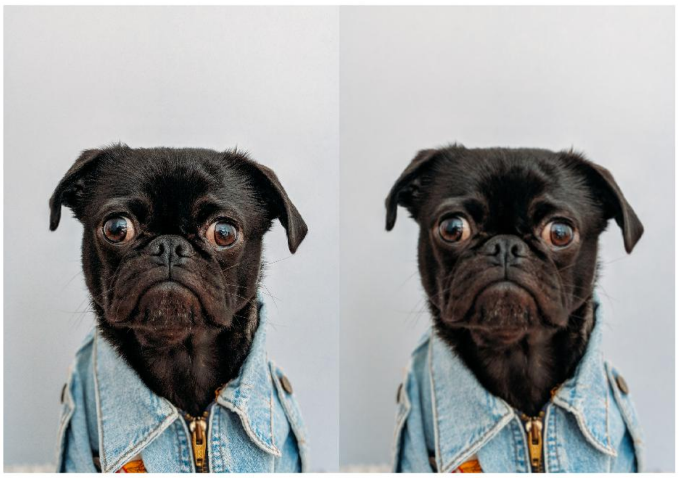


Figure 1.4: Visualization of a Filter Kernel Application

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.

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

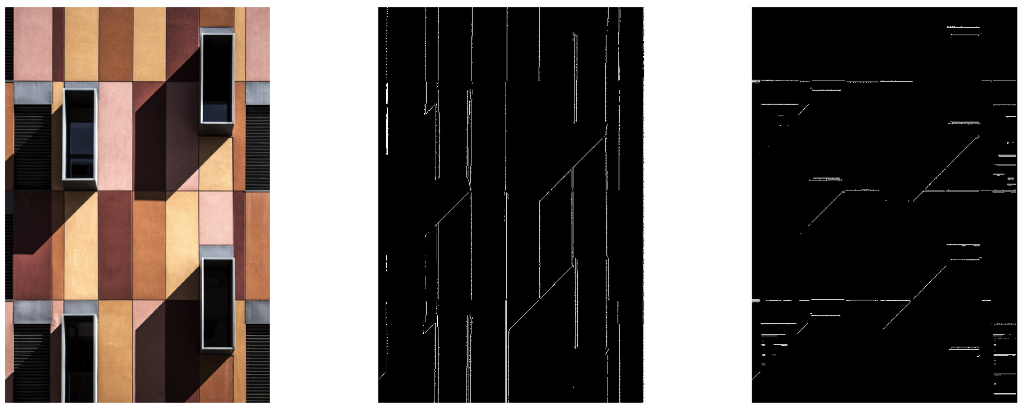
(a) Horizontal Sobel Filter Mask

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

(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.

$$\underset{F(x,y)}{\begin{array}{|c|c|c|c|c|}\hline 0&0&0&0&0\\\hline 0&0&0&0&0\\\hline 0&0&1&0&0\\\hline 0&0&0&0&0\\\hline 0&0&0&0&0\\\hline\end{array}} \otimes \underset{H(u,v)}{\begin{array}{|c|c|c|}\hline \mathrm{a}&\mathrm{b}&\mathrm{c}\\\hline \mathrm{d}&\mathrm{e}&\mathrm{f}\\\hline \mathrm{g}&\mathrm{h}&\mathrm{i}\\\hline\end{array}} = \underset{G(x,y)}{\begin{array}{|c|c|c|c|c|}\hline 0&0&0&0&0\\\hline 0&\mathbf{i}&\mathbf{h}&\mathbf{g}&0\\\hline 0&\mathbf{f}&\mathbf{e}&\mathbf{d}&0\\\hline 0&\mathbf{c}&\mathbf{b}&\mathbf{a}&0\\\hline 0&0&0&0&0\\\hline\end{array}}$$

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,

$$g = f * h$$

$$\underset{F(x,y)}{\begin{array}{|c|c|c|c|c|}\hline 0&0&0&0&0\\\hline 0&0&0&0&0\\\hline 0&0&1&0&0\\\hline 0&0&0&0&0\\\hline 0&0&0&0&0\\\hline\end{array}} * \underset{H(u,v)}{\begin{array}{|c|c|c|}\hline \mathrm{a}&\mathrm{b}&\mathrm{c}\\\hline \mathrm{d}&\mathrm{e}&\mathrm{f}\\\hline \mathrm{g}&\mathrm{h}&\mathrm{i}\\\hline\end{array}} = \underset{G(x,y)}{\begin{array}{|c|c|c|c|c|}\hline 0&0&0&0&0\\\hline 0&\mathbf{a}&\mathbf{b}&\mathbf{c}&0\\\hline 0&\mathbf{d}&\mathbf{e}&\mathbf{f}&0\\\hline 0&\mathbf{g}&\mathbf{h}&\mathbf{i}&0\\\hline 0&0&0&0&0\\\hline\end{array}}$$

Figure 1.9: Convolution of a filter and an image.

1.3 Clustering

Clustering is a method segmenting an image into classes which are disjoint sets. This is useful as each class can have a semantic label that converys useful information about an image.

[SHOW SCATTER PLOT OF CLUSERED DATA]

An entity's (for example a pixel) membership to a cluster is justified by it meeting some criteria that along with the rest of the cluster's members. A simple example of this is a luster defined by its location in an image, all pixels in that region would belong to that cluster. This is a very simple criteria and often more sophisticated prerequisites must be met in order to classify a pixel's cluster. Entities like pixels or a neighbourhood of pixels' characteristics are represented in something called a feature vector.

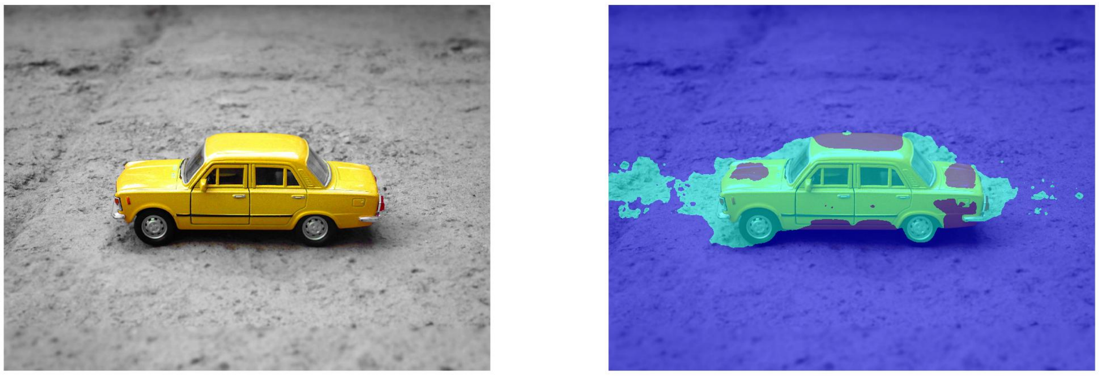

(a) Image by Gustavo, Upsplash. (b) Segmentation of car from background.

Figure 1.10: Segmentation of toy cars using K-Means Clustering.

1.3.1 K-Means

There are many methods by which to segment an image with clustering but K-Means is very popular for its simplicity and speed. K-means produces K clusters implemented as follows

1. Place K points on the image randomly or with some educated guess. These points are the initial centroids for the K clusters.
2. Assign all data points to their nearest centroid.
3. Update the centroids' positions as the mean of all data points in that class.
4. Repeat steps 2 and 3 until cluster centroids no longer move (converge).

K-means is guided by trying to limit the variance within each class. Variance is described by the expression

$$V = \sum_{i=1}^{k} \sum_{x_j \in c_i} (x_j - \mu_i)^2 \tag{1.3}$$

Where x_j are the data vectors, c_i are the classes that the data point belong to and μ_i are the class centroids.

Because the K-Means algorithm is fast and the success of the segmentation depends on the initial placement of centroids, often the algorithm is performed many times with different initial conditions and the instance that yields the best results is used.

[SHOW EXAMPLE OF K-MEANS CLUSTERING BEING USED]

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.