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

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 a broad field of study and so only concepts relevant to the system are detailed.

1.1 Digital Images

Unlike its analog predecessor the digital image consists of a discrete number of discrete valued data points called pixels. An image's resolution is defined how many pixels are in it and a pixel's value determines its colour. This value is the pixel's *intensity* and lies within a spectrum of values, for example an 8-bit image can have pixel values $[0, 255]$.

Digital images are often represented as a 2D array of values where each cell corresponds to a pixel value. In software this is how digital images are stored and manipulated. The image's array is $M\ pixels \times N\ pixels$ in size. See FIGURE XXX

[SHOW EXAMPLE ARRAY REPRESENTATION]

This representation allows pixels to be referenced by their relative position in the M and N directions. Furthermore, M and N may be substituted for something more familiar like

the x and y-axis of a Cartesian plane. In fact, a digital image is just a two dimensional function. That is, each pixel's position may be described by a coordinate (x,y) where we find the pixel's value (intensity). That is,

$$z = f(x, y)$$

This means that operations that can be applied to two dimensional functions can be applied to a digital image. With these operations all sorts of useful things may be done. Images can even be modelled as 3D objects as a consequence.

[SHOW ARRAY REPRESENTATION and 3D OBJECT REPRESENTATION]

1.2   Linear Filtering

There are many useful operations that may be applied to a digital image to augment or extract information from it. *Linear filters* are a subset of these operations. They operate on a *neighbourhood* of pixels in an image. The size of the neighbourhood to take as input is determined by the filter's size.

Perhaps the simplest linear filter is the weighted average filter, also known as the 'box filter'. This filter finds the weighted average of the *target pixel*, where the target is the pixel at the center of the filter's array of weightings (filter coefficients). The array of coefficients is also referred to as the filter's *kernel* or *mask*. For example, a box filter of size three that simply outputs the average of the target pixels neighbourhood would look like this,

[SHOW BOX FILTER MASK MATRIX]

Here, the weights are evenly distributed and equal to the chosen weight divided by the sum of all weights. By passing this filter over an entire image the sharpness of the image

is reduced as all the values are averages of their neighbourhood, giving a smoothing or blurring effect.

[SHOW OUTCOME OF APPLYING BOX FILTER TO IMAGE]

The application of a linear filter $h(u, v)$ to an entire image $f(i, j)$, where $h(0, 0)$ is the center of the filter mask and $f(0, 0)$ is the target pixel beneath the masks center, may be described as follows

$$g(i, j) = \sum_{u=-k}^{k} \sum_{v=-l}^{l} f(i + u, j + v) h(u, v) \tag{1.1}$$

where the dimensions of the filter kernel are $2k + 1 \times 2l + 1$. Filter kernels are nearly always odd dimensioned so as to have a center. This is because an asymmetric kernel (even numbered dimensions) will result in information being distorted because the output from the filter will be weighted unevenly by being placed in a reference pixel more one to one side than another.

[SHOW RESULT OF USING ASYMMETRICAL KERNEL]

1.2.1 Correlation

Linear filtering may be notated more concisely by the *correlation* operator.

$$g = f \otimes h$$

This operation measures the similarity between two signals (functions). We can think of both digital images and linear filters as signals. Performing correlation between them will yield an image where the highest values (pixel intensities) correspond to where the image and filter were most similar[1].

[SHOW EXAMPLE WITH SHARPENING FILTER]

This operation is *shift invariant*, which means that it does the same thing no matter where in an image it is applied. Therefore, correlation operations obey both the superposition principle

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

and the shift invariance principle

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

This operation 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.

[SHOW EXAMPLE OF HOW CORRELATION FLIPS INPUT FOR OUTPUT]

1.2.2   Convolution

Convolution is also a linear operation that is shift invariant. It is very similar to correlation except that the output signal from a convolution operation is not flipped relative to the input signal. It is described mathematically by the expression,

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

We see that this is similar to equation 1.1 but that the filter $h(i,j)$ is the entity being shifted, furthermore it is flipped (note the negative signs), this results in the output's

orientation being correct. Convolution may be notated with the following operator

$$g = f * h$$

[SHOW EXAMPLE OF CONVOLUTION NOT FLIPPING OUTPUT]

1.3 Clustering

This is a method of separating an image in disjoint sets (clusters). This is useful as we can semantically label different sections of an image according to their similarity.

[SHOW EXAMPLE OF IMAGE SEGMENTED BY CLUSTERING]

An entitys (for example a pixel) membership to a cluster is justified by it meeting some criteria that along with the rest of the clusters 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.

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

$$V = \sum_{i=1}^{k} \sum_{x_j \in c_i} (x_j - \mu_i)^2 \tag{1.2}$$

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]

1.4   Gaussian Mixture Model

1.5   Morphology

1.6   Contours

1.7   Centroid Tracking

Chapter 2

Design

Containing a comprehensive description of the design chosen, how it addresses the problem, and why it is designed the way it is.

Chapter 3

Implementation

Containing a comprehensive description of the implementation of your software, including the language(s) and platform chosen, problems encountered, any changes made to the design as a result of the implementation, etc.

Chapter 4

Summary and Reflections

Including a discussion of results in a wider context (considering other work).

4.1 Project management

Covering the tasks as a part of your work plan and progress as well as how time and resources are managed.

4.2 Contributions and reflections

Providing the details of your achievements and contributions including innovation, creativity and novelty (if there is any) as well as a personal reflection on the plan and your experience of the project (a critical appraisal of how the project went).

Bibliography

[1] SAHOO, S. Deciding optimal kernel size for cnn.

Appendix A

User Manuals

Appendix B

User Evaluation Questionnaire