

The entire code repository can be found at https://github.com/dgsaf/comp3007_assignment.

Contents

1	Introduction	2
2	Problem Analysis	2
2.1	Task 1	3
2.2	Task 2	3
3	Implementation	6
3.1	Overview	6
3.2	Mathematical Details	6
3.3	Detection	8
3.4	Classification by k -Nearest Neighbours	9
3.5	Task 1	9
3.6	Task 2	9
4	Validation Performance	9
4.1	Task 1	9
4.2	Task 2	9
5	Conclusion	9
A	Source Code	10

List of Figures

1	Task 1 - Regular Cases	3
2	Task 1 - Edge Cases	4
3	Task 2 - Regular Cases	5
4	Task 2 - Edge Cases	5

List of Tables

1 Introduction

2 Problem Analysis

We are concerned with the detection and classification of two similar but distinct types of signs. The properties specific to each type of signage are discussed in [subsection 2.1](#) and [subsection 2.2](#). Here, we remark on the properties common to both of them, which are numerous and allow for a broadly unified approach to their detection and classification.

Each sign consists of a black background with white characters (digits and directional arrows) forming the foreground of the sign. The black background is rectangular in shape, and bounds the characters regions. Furthermore, the contrast between the black background and the white foreground is strong across all colour channels and in grayscale.

The digits are uniform in their construction, being printed in a monospace font with fixed height and allocated a fixed width; although each digit does not necessarily fully extend across its allocated width - as can be seen for the digit 1. Likewise, the directional arrows are also uniform in their construction but their dimensions are distinct from those of the digits - noticeably having a smaller height than the digits.

The digits do not overlap with each other (unless subject to camera artifacts/blurring) but are closely adjacent to each other, and regularly spaced. Each sign contains at least one sequence of three digits, which are ideally surrounded by a black sub-region of the sign; although for task 2, the gap between the leftmost digit and the edge of the sign can be very small.

In summary, the following properties of the signs are almost always observed, providing a robust foundation for their detection:

- The uniformly white colour of the characters;
- The uniformly black background of the sign;
- The strong contrast between the white characters and the black background across all red, green, and blue channels as well as in grayscale;
- The uniform height of the digits;
- The uniform spacing of the characters horizontally;
- The height of the digits exceeding the width;
- The uniform dimensions of the arrows;
- The presence of a chain of exactly three digits;
- A sizeable sub-region of the sign above and below the chain of digits, which is uniformly black.

However, the assumption of the above properties may be invalidated by any one of the (non-exhaustive) list of conditions:

- The presence of strong radial, or motion blurring which may cause character to overlap;
- The presence of strong shadowns, glare, or other variations in lighting conditions across the character;

- The presence of a foreign object, such as a sticker or a mark, on the sign (violating the assumptions on uniformity of layout) or across the characters (violating the assumptions on uniformity of characters).

It should also be noted that often there are other monospace white characters present near the sign - typically letters forming the name of the location marked by the sign. These characters can be distinguished from the digits (and arrows) by the fact that they almost always form chains of more than three characters, and presented on their own signs which almost never have a black background.

2.1 Task 1

Further refinement of the problem analysis is possible for task 1. We are concerned with finding only one chain of three digits, and we are not concerned with directional arrows at all. Furthermore, there exists a sizeable gap between the chain of digits and the edge of the sign both horizontally and vertically.

Bricks and other objects which are of approximately uniform dimensions and uniformly spaced are regularly present in images for task 1 - hence, when exploiting the uniformity of the layout of the digits, care must be taken to avoid or handle the false detection of these other objects.

Examples of the images expected for task 1, which provide the justification for the assumptions made on the properties of the sign are shown in [Figure 1](#). Examples of images which may be encountered for task 1, but which challenge the assumptions made on the properties of the sign are shown in [Figure 2](#).



Figure 1: Examples of training images for task 1, which exhibit the described distinguishing properties of the signs and digits.

2.2 Task 2

Similarly, further refinement of the problem analysis is possible for task 2. We are concerned with finding a variable number of chains of three digits, each of which has an associated directional arrow to their right. For each chain and their associated directional arrow, the geometric centres of the digits are arrow are collinear; hence we say that they form a line. Each line is vertically spaced (by a distance just larger than the height of the digits) and the set of digits and arrows from each line thus forms a grid-like layout.

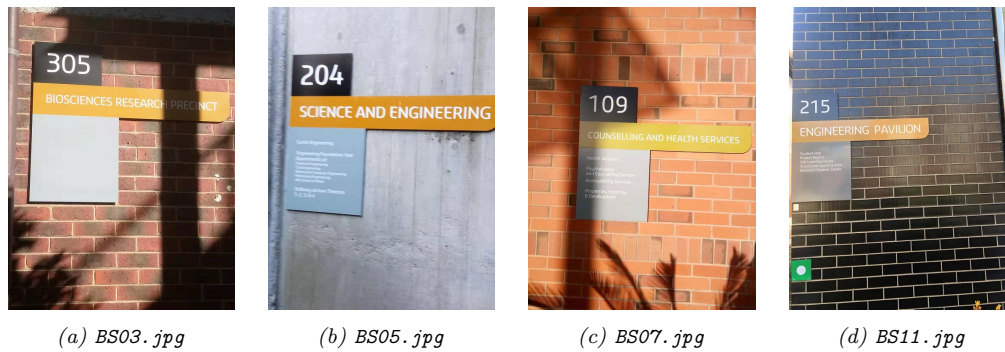


Figure 2: Examples of training images for task 1, which may challenge the assumptions made on the signs and thus necessitate the development of a robust detection algorithm. Sharp shadows, motion blur and monochromatic periodic objects, which may interfere with digit detection, can be observed.

A challenging aspect of these signs, compared to those for task 1, is that the gap between the leftmost digit on each line and the edge of the sign can be very small. In cases where an object, of similar colour/intensity to the white characters, is behind the sign but in a similar region of the image to the leftmost digit of a line, the digit and the object may prove difficult to distinguish as separate regions.

Another difficulty is that the number of lines in any particular sign is not known beforehand. However, there is a benefit in having multiple lines per sign as it provides a mechanism, given at least one fully known line or even partial information from multiple lines, to recover an expected location for characters which fail to be detected initially.

Similar to task 1, the scene environment can be expected to be quite noisy, with periodically repeating objects such as brickwork present, as well as other objects with many sharp edges, such as plants and trees. It also seen that foreign objects, such as stickers and marks, may be present on the signs, and that other objects, such as plants, may occlude the sign itself.

Examples of the images expected for task 2, which provide the justification for the assumptions made on the properties of the sign are shown in [Figure 3](#). Examples of images which may be encountered for task 2, but which challenge the assumptions made on the properties of the sign are shown in [Figure 4](#).

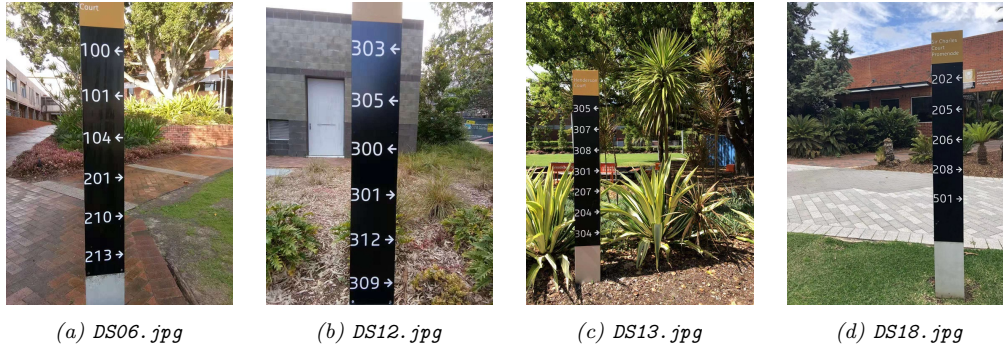


Figure 3: Examples of training images for task 2, which exhibit the described distinguishing properties of the signs, and their digits and arrows.

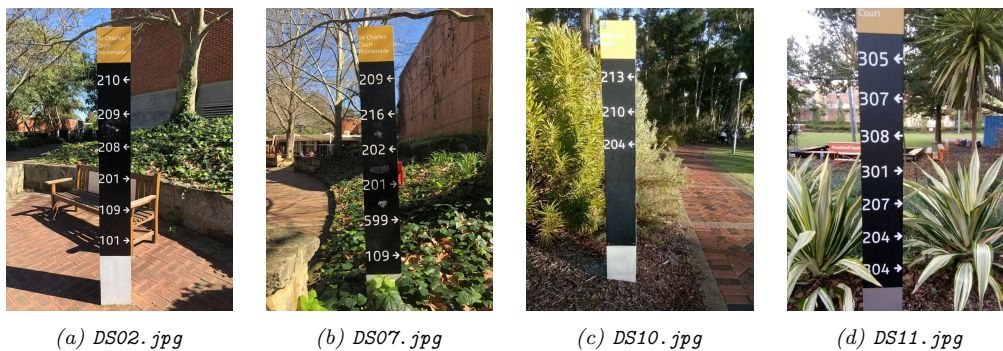


Figure 4: Examples of training images for task 2, which may challenge the assumptions made on the signs and thus necessitate the development of a robust detection algorithm. Characters overlapping with similar regions outside the sign, foreign objects on the sign, variable lighting conditions can all be observed.

3 Implementation

3.1 Overview

We make use of the commonality of the properties of the signs that are to be detected in task 1 and task 2 by designing a detection algorithm which can be adapted with minor changes for each task. The specific detection algorithms for each task are discussed in [subsection 3.5](#) and [subsection 3.6](#). Furthermore, as the digits to be classified in each task are of the same style, we also design a classification algorithm which is suitable for both tasks. Naturally, a separate classifier will be needed for the directional arrows, however we use the k -Nearest Neighbours (k -NN) algorithm for classifying both the digits and the arrows. Hence, we discuss the problem of classification, for both types of characters, in a general manner in [subsection 3.4](#).

To detect the characters (digits and arrows) we utilise the method of Maximally Stable Extremal Regions (MSER) - first developed by [Matas et al., 2004] and we present our formalism in a similar manner to theirs. In this method, a number of regions, which are extremal in terms of light intensity and stable in the sense that the region boundary changes little under a varying threshold, are detected. With suitably chosen parameters for the MSER algorithm, the regions of interest, corresponding to the characters on the sign, can be reliably detected - but must be distinguished from the other regions detected which are not of interest. Using known properties of the characters (such as dimension, aspect ratio, bounding box fill) we are able to refine the set of regions. We then examine the geometric relationships between the regions to determine which pairs of regions are: similar in dimension, and possibly collinear. This results in a directed graph of regions, with paths in this graph constituting possible lines of similar regions (a property that the sequences of digits are known to exhibit).

3.2 Mathematical Details

Image

We define an image I , with n channels and of width w and height h , to be a mapping

$$I : D \rightarrow S : (x, y) \mapsto I(x, y) = (I_1(x, y), \dots, I_n(x, y)), \quad (1)$$

where the domain $D \subset \mathbb{N}^2$ is of the form $D = \{0, \dots, w-1\} \times \{0, \dots, h-1\}$, and S is the codomain. Constraints can be specified on S , but for our purposes it suffices to suppose that either $S \subseteq \{0, \dots, 255\}^n$ or $S \subset \mathbb{R}^n$.

Thresholded Image

Suppose $I : D \rightarrow S \subset \mathbb{R}$ is a single-channel image. For each $t \in S$, we define the Boolean-valued image I_t to be of the form

$$I_t : D \rightarrow \mathbb{B} : (x, y) \mapsto \begin{cases} 0 & \text{if } I(x, y) \leq t \\ 1 & \text{if } I(x, y) > t \end{cases}, \quad (2)$$

and we say that I_t is a thresholded image.

Bounding Box

Suppose D is the domain of an image $I : D \rightarrow S$. For all $R \subseteq D$, we define the bounding box $B(R)$ of R to be

$$B(R) = [x_R, x_R + w_R] \times [y_R, y_R + h_R] \quad \text{such that} \quad R \subseteq B(R) \quad (3)$$

and where $w_R, h_R \in \mathbb{N}$ are minimal.

Connectedness

Suppose D is the domain of an image. An adjacency relation A on D is a Boolean-valued mapping

$$A : D \times D \rightarrow \mathbb{B} : (p, q) \mapsto A(p, q), \quad (4)$$

which indicates if the two points of the domain are considered to be adjacent. For any $p \in D$, we may define the neighbourhood $N(p)$ of p to be the set of all points which are adjacent to it; that is,

$$N(p) = \{q \in D \mid A(p, q)\}. \quad (5)$$

For any $p, q \in D$, we say that p and q are connected if there exists a finite sequence $(\rho_k)_{1 \leq k \leq n}$ in D such that

$$A(p, \rho_1) \wedge A(\rho_1, \rho_2) \wedge \cdots \wedge A(\rho_{n-1}, \rho_n) \wedge A(\rho_n, q) = 1. \quad (6)$$

In the case where the adjacency relation A is symmetric, then connectedness defines an equivalency relation; whence we may write, for any connected $p, q \in D$ that $p \sim q$.

Note that we are primarily concerned with the adjacency relations associated with the Von Neumann neighbourhood (4-connectivity)

$$N_4(p) = \{p + n \in D \mid n \in \{(0, 1), (1, 0), (0, -1), (-1, 0)\}\}, \quad (7)$$

and the Moore neighbourhood (8-connectivity)

$$N_8(p) = \{p + n \in D \mid n \in \{-1, 0, 1\} \times \{-1, 0, 1\} \setminus (0, 0)\}. \quad (8)$$

Region

Suppose D is the domain of an image $I : D \rightarrow S$ and let $A : D \times D \rightarrow \mathbb{B}$ be a symmetric adjacency relation on D . We say that $R \subseteq D$ is a region if every element of R is connected to every other element of R ; that is,

$$p, q \in R \implies p \sim q. \quad (9)$$

We define the (inner) boundary ∂R of a region R to be subset of points of R which are also connected to at least one point not in R ; that is,

$$\partial R = \{p \in R \mid \exists q \in D \setminus R : A(p, q)\}. \quad (10)$$

We define the outer boundary ΔR of a region R to be the set of points of D which do not belong to R but are adjacent to a point of R ; that is,

$$\Delta R = \{p \in D \setminus R \mid \exists q \in R : A(p, q)\}. \quad (11)$$

Maximally Stable Extremal Region (MSER)

Suppose $I : D \rightarrow S \subset \mathbb{R}$ is a single-channel image, and suppose $A : D \times D \rightarrow \mathbb{B}$ is the (symmetric) adjacency relation associated with either the Von Neumann neighbourhood or the Moore neighbourhood. Suppose $R \subseteq D$ is a region. We say that R is a minimal region if for all $p \in R$ and $q \in \Delta R$ we have $I(p) < I(q)$; which is equivalently written as the requirement that

$$\max_{p \in R} I(p) < \min_{q \in \Delta R} I(q). \quad (12)$$

Similarly, we say that R is a maximal region if for all $p \in R$ and $q \in \Delta R$ we have $I(p) > I(q)$; which is equivalently written as the requirement that

$$\min_{p \in R} I(p) > \max_{q \in \Delta R} I(q). \quad (13)$$

We say that R is an extremal region if it is either a minimal or maximal region.

The formulation of extremal regions in terms of the minimal and maximal intensity values of the image permits the usage of thresholding. Suppose that R is an extremal region, and suppose that $t \in S$. Consider the thresholded region R_t , defined by

$$R_t = \{p \in R \mid I(p) < t\}, \quad (14)$$

which is itself an extremal region, and for which we have that

$$\max_{p \in R_t} I(p) < t. \quad (15)$$

We note that $R_t \subseteq R$ for all $t \in S$. We also note that when $t_1 \leq t_2$, we have that $R_{t_1} \subseteq R_{t_2}$; that is, the thresholded regions form an increasing (by set inclusion) sequence of subsets of R . For any increasing chain $t_1 < \dots < t_n$ in S , we have

$$\emptyset \subseteq R_{t_1} \subseteq \dots \subseteq R_{t_n} \subseteq R. \quad (16)$$

In the MSER approach, the stability of an extremal region R is measured by examining the change in the cardinality of R_t with the change in the threshold t . That is, for a particular threshold $t \in S$ and threshold step $\delta \in S$, such that $t - \delta, t + \delta \in S$, the rate of growth of the extremal region R is given by

$$G_\delta(R; t) = \frac{|R_{t+\delta} \setminus R_{t-\delta}|}{|R_t|}. \quad (17)$$

An extremal region R_{t_0} is then said to be maximally stable if $G_\delta(R; t)$ has a local minimum at $t = t_0$. Such a thresholded region experiences minimal change (in cardinality) when the threshold t_0 is increased/decreased by δ .

3.3 Detection

The problem of detection is accomplished by using the MSER method to detect the character regions present on the sign. We suggest that this technique is suitable for our purpose, as the characters are uniformly white and contrast well against the surrounding uniformly black bounding area of the signs. Furthermore, in most cases the characters are well separated from other characters and

the edge of the sign (except notably for the leftmost digits in task 2). Hence, these characters should form extremal regions which are relatively stable under affine transformations, and when the image is underexposed or overexposed. Theoretically, this technique may be susceptible to extreme variations in lighting conditions, such as strong shadows or glare, which form discontinuous non-monotonic changes in light intensity. However, in testing, this technique has proved tolerant to moderate shadows and glare, so we do not expect this to be a major issue.

3.4 Classification by k -Nearest Neighbours

3.5 Task 1

3.6 Task 2

4 Validation Performance

4.1 Task 1

4.2 Task 2

5 Conclusion

References

[Matas et al., 2004] Matas, J., Chum, O., Urban, M., and Pajdla, T. (2004). Robust wide-baseline stereo from maximally stable extremal regions. *Image and Vision Computing*, 22(10):761–767. British Machine Vision Computing 2002.

A Source Code