参赛学生姓名：张景涵　田桢干

中学：南京外国语学校

省份：江 苏

国家/地区：中 国

指导老师姓名：张 川

指导老师单位：东南大学

论文题目：A Novel Crosswalk Traffic Light Detection Algorithm Resolving Multi-Light Interference for the Visually Impaired

A Novel Crosswalk Traffic Light Detection Algorithm Resolving Multi-Light Interference for the Visually Impaired

Jinghan Zhang, Zhengan Tian

Nanjing Foreign Language School, Nanjing, China

Abstract: This paper proposes a novel algorithm for detecting crosswalk traffic lights, specifically designed to aid visually impaired pedestrians. The algorithm addresses the challenge of multi-light interference by associating crosswalks with their corresponding traffic lights, significantly improving detection accuracy. Additionally, the algorithm incorporates squeeze-and-excitation (SE) and convolutional block attention module (CBAM) mechanisms to enhance the detection of small objects and crosswalks under various conditions. Experimental results show that our algorithm achieves a detection accuracy of 97.5% with score weight (0.5,0.5), substantially outperforming the 92.0% accuracy of the YOLOv5 combined with the maximum pixel detection method. Sensitivity analysis also reveals that our algorithm has strong robustness to noise. This effectively improves the safety of visually impaired individuals while crossing crosswalks.

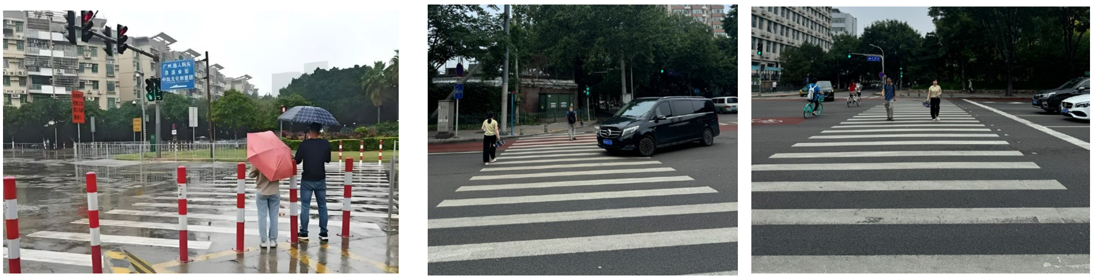
Keyword: Crosswalk traffic light, Crosswalk, Object detection, Convolutional neural network, YOLOv5, Visually impaired assistance

1. Introduction

The visually impaired community is easily overlooked. In their daily lives, they often encounter many inconveniences. For example, blind pathways are often obstructed, and most crosswalk traffic lights lack indicators for the visually impaired. In our communications with visually impaired friends, we learned that it is different from our former assumptions, most of them can use smartphones quite well through voice assistance. They can utilize functions like taking photos and even sharing pictures on social media. Therefore, we decided to utilize the camera function of smartphones to develop an algorithm for detecting crosswalk traffic lights to help visually impaired individuals safely cross the street.

In recent years, since the demand for autonomous vehicles and smart city applications has grown significantly, traffic light detection and recognition technologies have advanced quickly. Traditional traffic lights detection approaches mainly rely on classic computer vision techniques, such as color segmentation, edge detection, and shape recognition, to identify traffic lights. However, different lighting conditions, obstruction, and weather conditions frequently make these approaches less reliable in practical scenarios. Recently, methods combined with deep learning, especially convolutional neural networks (CNNs)[1], have been proposed to improve the accuracy and robustness of traffic light detection. CNN-based object detection algorithms are generally categorized into two types: two-stage detection algorithms and one-stage detection algorithms. Two-stage algorithms, such as R-CNN[2] series, first input the image into a convolutional neural network (CNN) for feature extraction and generate a sparse set of candidate regions. Then, a second CNN is used to perform targeted feature extraction on these candidate regions. Finally, the system perform classification. One-stage detection algorithms do not require generating pre-selected regions that might contain objects, such as YOLO[3] series and RetinaNet[4]. Instead, they divide the image into a grid (e.g., 10×10 cells) and detect objects directly within each grid cell, significantly reducing the processing time. Among them, YOLOv5[5] is often used for detecting traffic signals for its fast detection speed and high accuracy.

However, most current traffic light detection focuses on vehicle traffic lights, paying very little attention to crosswalk traffic light. Unlike vehicle traffic light detection, crosswalk traffic light detection not only needs to identify smaller targets, but also needs to detect the crosswalk traffic light from multiple traffic lights in one image. In figure 1, there are multiple traffic lights in each image. For example, in figure 1(a), the crosswalk traffic light is located to the left of the blue umbrella and is quite small and difficult to identify. The other two vehicle traffic lights in this image are more obvious, and because the colors are different, it is very easy to misjudge. At present, there is no research work that simultaneously considers these two challenges: selecting the right light from multi lights and small targets detection. Such as in references [6-8], although the crosswalk traffic lights are identified, they are not distinguished from the traffic lights of vehicles or other crosswalks light in the same image. As a result, these methods may lead to detection errors and cause safety problems.



(a) (b) (c)

Figure1. Images of crosswalk traffic light

In this paper, we propose a novel crosswalk traffic light detection algorithm with anti-multi-light interference capability for the visually impaired. When there are multiple traffic lights in an image, the crosswalk traffic light will be detected based on the crosswalk location to resolve multi-light interference. At the same time, two attention mechanisms are introduced. Squeeze-and-excitation (SE) [9] is used for detecting small targets, convolutional block attention module (CBAM)[10] is used for detecting crosswalk. The main contributions of this paper are as follows:

1. To solve the problem of misidentification caused by multiple traffic lights, the designed algorithm associates crosswalk traffic lights with the corresponding crosswalk location. It effectively improves the detection accuracy and makes crossing the road safer for the visually impaired.
2. To improve the detection rate of traffic lights, the SE model is introduced after SPPF block of YOLOv5. By adaptively adjusting the importance of each feature channel, the SE module greatly enhances traffic light detection, which belongs to the category of small targets detection.
3. To detect the crosswalk, CBAM is added before each C3 block at the neck of YOLOv5 and after SPPF block at the backbone of YOLOv5. By introducing channel attention and spatial attention, the model can filter and enhance features before convolution, resulting in better detection accuracy.

4. To ensure the practicality and applicability of our work, we took many images of crosswalk traffic lights from the perspective of pedestrians and labeled them to establish a dataset. This dataset is intended to serve as a valuable resource for future researchers.

The remainder of this article is organized as follows. Section 2 provides the preliminaries of this work. The propose crosswalk traffic light detection algorithm is detailed in Section 3. Section 4 presents the experimental results and performance analysis. Finally, section 5 concludes the article and gives future research directions.

2. Preliminaries

2.1 CNN

In image classification tasks, a convolutional neural network is composed of convolutional layers, pooling layers, and fully connected layers. It performs classification by using a structure with multiple connected layers. A typical structure of CNN is shown in figure 2.

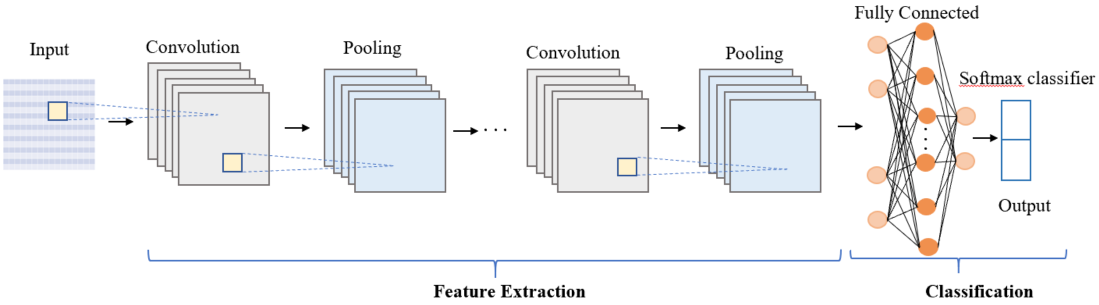


Figure 2. The structure of CNN

The primary function of the convolutional layer is to extract features presented in the previous layer. It uses different sizes of convolutional kernels (like 2×2 or 3×3) to capture local patterns. The convolutional kernel operates on the input image or feature map, and the results are passed through an activation function to enhance nonlinearity. The activation function is placed at the end or between layers of neural networks and transforms the input to keep its values within a manageable range. Common activation functions include the Sigmoid function, ReLU function, etc.. This process can be expressed as

$$C^l = f(A^l * W^l + B^l)$$

where C^l is the output of l layer, A^l is the input of l layer, W^l is the learnable weight vector, B^l is the bias vector, $f(\ )$ is the activation function, and * is the convolutional operation.

In CNN, a convolutional layer is followed by a pooling layer. Pooling layer, also known as downsampling layer, is used to reduce the size of the input feature map and increase the receptive field of the following convolutional layers, reducing overfitting in the network. A pooling operator integrates the data from a small area (like a rectangle) into a single value. The most popular pooling methods are max pooling and average pooling. Figure 3 presents the process of convolution and pooling.

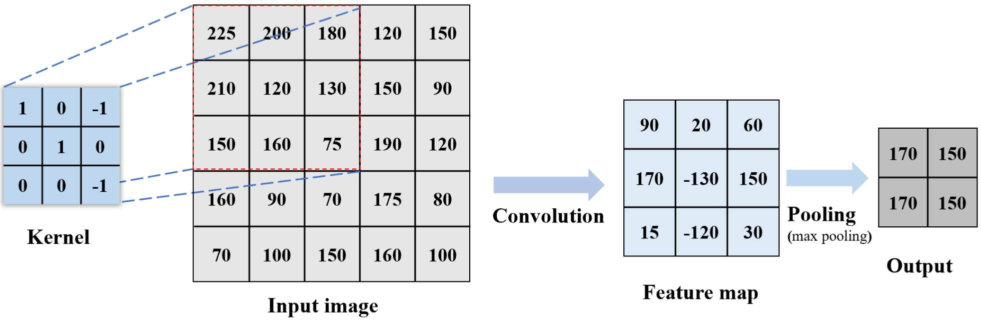


Figure 3. Example of the process of convolution and pooling (max pooling)

The fully-connected layer synthesizes the features into the final classification. In the fully-connected layer, each neuron is connected to all the input neurons from the previous layer, with connections governed by weights and biases. It can be expressed as:

$$y_j = f\left(\sum_i (w_{i,j}.x_i) + b_j\right)$$

where $w_{i,j}$ is the weight between neuron i and neuron j, x_i is the input from neuron i, y_j is the output of neuron j, b_j is the bias for neuron j, and $f(\ )$ is also the activation function.

At the end, the softmax function is applied in the output layer to normalize raw scores into a probability distribution, ensuring that the sum of probabilities across all classes equals 1.

2.2 Evaluation metrics

Confusion matrix is a table used to evaluate the performance of a classification model by comparing its predicted labels with the actual labels, as shown in figure 4.

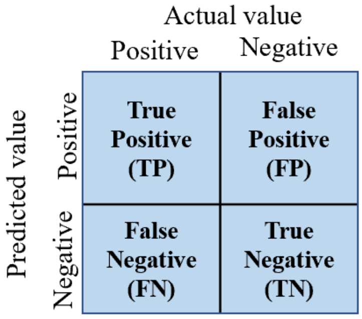


Figure 4. Confusion matrix

Here, true positive (TP) is the number of correctly predicted positive class by the model, true negative (TN) is the number of correctly predicted negative class by the model, false positive (FP) is the number of incorrectly predicted positive class by the model, and false negative (FN) is the number of incorrectly predicted negative class by the model. From confusion matrix, some evaluation metrics are defined.

Accuracy is the proportion of correct predictions made by the model out of all predictions:

$$\text{Accuracy}=\frac{TP+TN}{TP+TN+FP+FN}$$

Precision is the proportion of how many of the predicted positive results are actually positive:

$$\text{Precision}=\frac{TP}{TP+FP}$$

Recall is the proportion of actual positives that are correctly identified by the model:

$$\text{Recall}=\frac{TP}{TP+FN}$$

Specificity is the proportion of actual negative cases that are correctly identified by the model:

$$\text{Specificity}=\frac{TN}{TN+FP}$$

F1-Score is the harmonic mean of precision and recall, used when you want a balance between the two:

$$\text{F}_1\_\text{Score}=2\times\frac{\text{Precision}\times\text{Recall}}{\text{Precision+Recall}}$$

mAP is the average of the average precision (**AP**) across all classes, where AP is the area under the precision-recall curve for a given class:

$$\text{AP}=\int_0^1 p(r)dr$$

where $p(r)$ is the precision as a function of recall.

$$\text{mAP}=\frac{1}{N}\sum_{i=1}^{N}\text{AP}_i$$

where AP*i* is the average precision for class *i*.

Receiver operating characteristic (**ROC**) is a graphical representation used to assess the performance of a binary classification model. The ROC curve plots the true positive rate (TPR), also known as recall, against the false positive rate (FPR) at different threshold settings. It helps in visualizing how well a model distinguishes between two classes. FPR is calculated as:

$$\text{FPR}=\frac{FP}{FP+TN}$$

The area under the ROC curve (**AUC**) is a single value summarizing the performance of the model. An AUC of 1 indicates perfect performance, while an AUC of 0.5 suggests that the model performs no better than random distribution.

Among these metrics, precision focuses on how accurate the positive predictions are, i.e., of all the instances predicted as positive, how many are actually positive. Precision is important when false positives carry a high cost, such as in medical diagnoses or traffic light detection, where incorrect predictions could lead to harm or accidents.

3. Implementation

3.1 System Framework

To achieve better detection results, we took many photos from the perspective of pedestrians crossing the street. By analyzing the images, we summariz the characteristics of this type of images:

(1) Crosswalk traffic light always appears together with crosswalk;

(2) Multiple traffic lights often appear in one image;

(3) In some images, the crosswalk traffic light is smaller than other vehicle traffic lights;

(4) Photos can be taken from different angles.

For these characteristics, the basic YOLOv5 algorithm is not suitable for detecting crosswalk traffic lights in these images. This algorithm treats each object independently and may fail to understand that the presence of a crosswalk increases the likelihood that the nearby traffic light is the crosswalk traffic light. It might confuse crosswalk traffic lights with vehicle traffic lights, especially when multiple lightss are present, leading to false positives or negatives. YOLOv5 may not effectively detect small objects because its feature maps might not capture the necessary fine-grained details. In addition, unusual angles can distort object shapes and features, making detection more difficult.

Therefore, we propose a new algorithm for crosswalk traffic light detection. As shown in Figure 5, after inputting the image, the improved YOLOv5 algorithm with the SE attention module is first used to detect the crosswalk traffic light. If only one traffic light is detected, the detected traffic light will be sent to color detection module. If multiple traffic lights are detected in the image, the image is reprocessed into the improved YOLOv5 algorithm with the CBAM attention module to detect the crosswalk. Based on the detected crosswalk, a nearby area will be selected, and the crosswalk traffic light will be detected in the selected area. Finally, the detection results will be sent to color detection module.

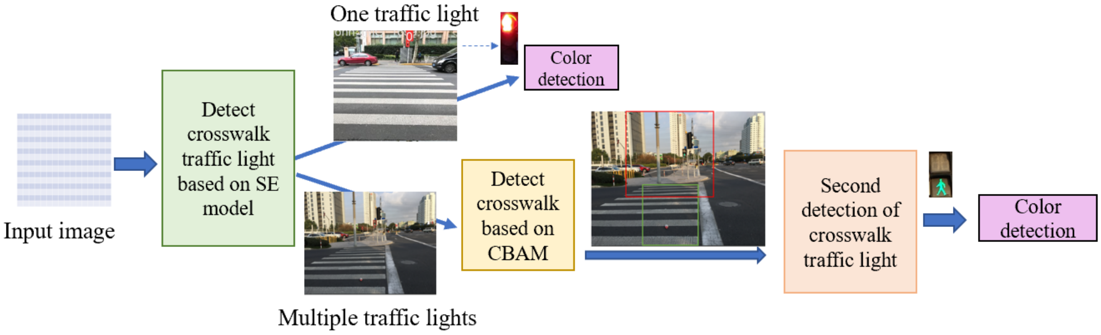

One traffic light
Color detection
Detect crosswalk traffic light based on SE model
Input image
Detect crosswalk based on CBAM
Second detection of crosswalk traffic light
Color detection
Multiple traffic lights

Figure. 5 System framework of crosswalk traffic light Detection algorithm resolving multi-light Interference

3.2 Detect crosswalk traffic light based on SE model

The traffic light detection requires the model to accurately identify small and specific objects. The SE module can adaptively assign weights to each channel, enhancing the attention on important feature channels. The SE module is added after the SPPF module of YOLOv5. By the squeeze operation, it compresses global information into a single value for each channel, and then reallocates the importance of each channel through a fully connected layer by excitation operation. This process can dynamically adjust the responses of individual channels based on global information, enhancing the characteristic information of small targets.

By the squeeze operation, global spatial information from each channel is compressed into a single value using global average pooling. This operation effectively squeezes the spatial dimensions of the feature map. This operation can be formulated as

$$z_c = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} x_c(i, j)$$

where $x_c(i,j)$ is the value at spatial position (i,j) of the c-th feature map, H and W are the height and width of the feature map, and z_c is the squeezed scalar for channel c.

Excitation operation is after squeezing operation and produces channel-wise weights that scale the importance of each channel. This operation passes the global pooled information through two FC (fully connected) layer with a non-linear activation function ReLU followed by a Sigmoid activation. This operation can be formulated as:

$$s_c = \sigma(W_2 \delta(W_1 z))$$

where W_1 and $W2$ are the weights of the two FC layer, δ is the ReLU activation function, σ is the Sigmoid activation function, and s_c is the final weight for the *c-th* channel.

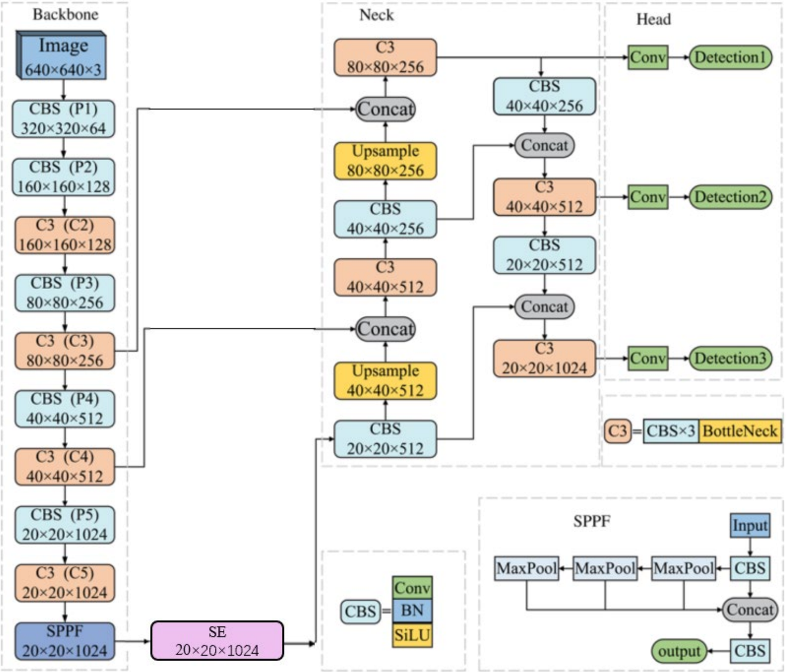


Figure 6. Improved YOLOv5 with SE model for crosswalk traffic light detection

The remaining blocks in Figure 6 are:

(1) Backbone

The backbone of the network is responsible for extracting features from the input image. It consists of the following components:

• Image Input: A 640x640 pixel image with 3 color channels (RGB).

• CBS Blocks: Convolutional layers followed by Batch Normalization and SiLU (Sigmoid Linear Unit) activation.

• C3 Blocks: Blocks consisting of three CBS blocks followed by a residual connection (skip connection).

• SPPF Blocks: Module that aggregates contextual information by using multiple max-pooling operations of different kernel sizes, which has an output size of 20 × 20 ×1024.

(2) Neck

The neck is responsible for fusing features from different scales (feature pyramid) and enhancing the model's ability to detect objects of different sizes.

• Upsample and Concat: Features from different layers are upsampled and concatenated with higher-resolution feature maps:

– The 20×20×1024 features from the SPPF block are concatenated with the 20 ×20×512 features from the C4 block, followed by a C3 block to output 20 ×20×1024 features.

– These are further upsampled to 40 × 40 × 512, concatenated with the 40 × 40×512 features from C3, and processed through another C3 block to output 40 ×40×512 features.

– This is again upsampled to 80 × 80 × 256, concatenated with 80 × 80 × 256 features from C3, and passed through a C3 block to produce 80 × 80 × 256 features.

(3) Head

The head consists of three detection layers that predict bounding boxes, objectness scores, and class probabilities at three different scales:

- Detection1: Operates on 80 ×80×256 features for detecting small objects.
- Detection2: Operates on 40×40×512 features for detecting medium-sized objects.
- Detection3: Operates on 20 ×20×1024 features for detecting large objects. Each detection layer consists of a convolutional layer that outputs predictions.

By adopting this improved YOLOv5 with SE model, the crosswalk light will be detected, but the other traffic lights will also be selected. Therefore, if only one traffic light in an image, the box of traffic light will be cropped, and the cropped image will be sent to color detection module, else will be sent to the "Detect crosswalk based on CBAM" module, as shown in figure 5.

3.3 Detect crosswalk based on CBAM

For images with multiple traffic lights, the crosswalk will be used to help locate the crosswalk traffic light to reslove multi-light interference. An improved YOLOv5 algorithm based on CBAM is designed to detect crosswalk.

CBAM is an attention mechanism that can be added to convolutional neural networks to enhance feature representation by focusing on important channels and spatial locations. We adopt CBMA to YOLOv5 for crosswalk detection to improve the detection accuracy for crosswalks by focusing on the most important features in the image. CBAM does this through two types of attention: channel attention and spatial attention.

Channel attention helps the model focus on the most important feature channels and uses global average pooling and global max pooling to figure out which channels are important. With channel attention, the model can give more weight to the channels that are most relevant to detecting crosswalks, ignoring unnecessary background noise. The channel attention can be formulated as

$$M_c(F) = \sigma(W_1(W_0(AvgPool(F))) + W_1(W_0(MaxPool(F))))$$

where F is the input feature map, AvgPool and MaxPool extract global features from each channel, W_0 and W_1 are weights in the network, σ is the Sigmoid function.

Spatial attention makes the model concentrate on the most important areas where crosswalk lines are located, especially when those lines might be partially hidden or difficult to see. The spatial attention can be formulated as

$$M_s(F) = \sigma(f^{7\times 7}([AcgPool(F); MaxPool(F)]))$$

where $f^{7\times 7}$ represents a convolution operation with the filter size of 7× 7, σ is the Sigmoid function, and $[AcgPool(F); MaxPool(F)]$ combines the results of average and max pooling.

Therefore, as shown in figure 7, we add CBAM after SPPF at the tail of the backbone, which enhances the network's ability to select important features across channels and spatial regions. The SPPF module has already helped to capture features at different scales, but adding CBAM after SPPF ensures that the most relevant scale-specific features are emphasized. This means the model is better at detecting crosswalks regardless of their sizes or orientations in the frame. We also add an CBAM module before each C3 block in the neck. This is particularly helpful in the neck of the YOLOv5 architecture, where features are aggregated from different layers (backbone and previous neck layers).

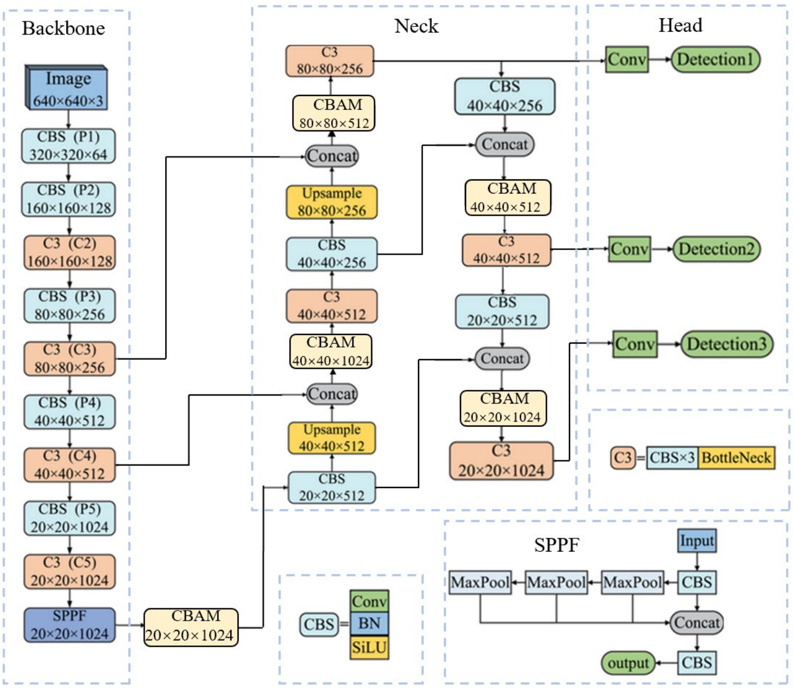


Figure 7.　Algorithm structure for crosswalk detection

Figure 8 is the detection results. We can see the crosswalks are marked with green boxes. By labeling the training set and refining the model, the detection results are restricted to crosswalks made up of horizontal lines, corresponding to the current street crossing. This helps avoid interference from crosswalks on other roads.

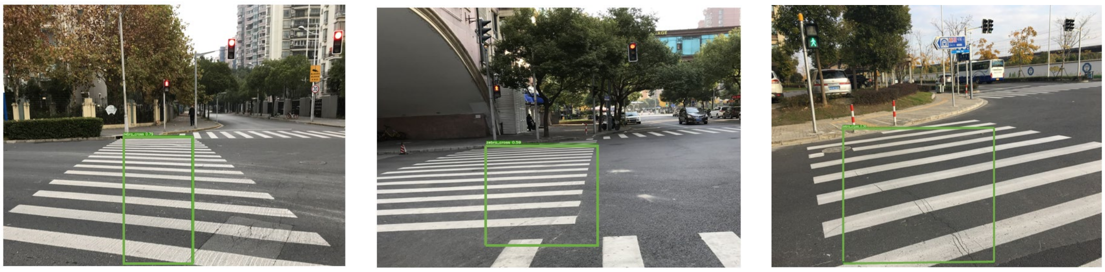
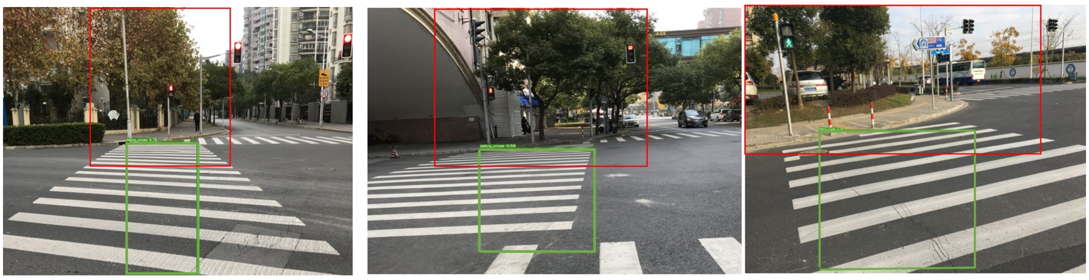


Figure 8. The images marked with crosswalk bounding boxes

3.4 Second detection of crosswalk traffic light

For situations with multiple traffic lights, after detecting the crosswalk, we will select a specific area around the crosswalk and perform the second detection of the crosswalk traffic light in this area. The detection method still uses the method in section 3.2. Choosing a detection area that takes into account the crosswalk locations is crucial. Analyzing the images marked with crosswalk bounding boxes which are the output of crosswalk detection, we find that the crosswalk lights are generally located near the crosswalk bounding boxes. Because of different camera angles, most traffic lights are positioned directly above or slightly diagonally above the top line of crosswalk bounding, but a few are located below, as shown in figure.8.

Therefore, we use the following cropping function to select the detection area. The cropping function expands the bounding box by *m*% horizontally, while the vertical cropping is adjusted to include the area from the top of the image to *n*% below the top boundary of the crosswalk's bounding box. Most of the target traffic lights are exactly above the crosswalk, but some are outside this range. The *m*% expansion in the horizontal direction is intended to broaden the search area for potential traffic lights around the detected crosswalk, which increases the likelihood of capturing targeted traffic lights that are not on the top of the bounding box. Expanding the bounding box downwards by *n*% beyond the top boundary of the crosswalk's bounding box allows us to account for variations in traffic light placement height. This small expansion ensures that even if the traffic lights are positioned a bit lower than the bottom of the crosswalk, they will still be captured within the cropped area.



(a) (b) (c)

Figure 9. The selected area (in red box) where *n*=50, *m*=20

If multiple traffic lights are still detected within this selected image, such as in figure 9 (c), we choose the most relevant traffic light using the following method. A score *S* is calculated for each detected traffic light based on two factors: D_i the normalized distance of the *i*-th traffic light's center to the midpoint of the upper boundary of the crosswalk's bounding box, and *A* is the normalized area of the traffic light's bounding box.

$$D_i = \frac{d_i^{\,2}}{d_{\max}^{\;\;2}} \qquad i \in (1, N)$$

where d_i is the distance of the *i*-th traffic light's center to the midpoint of the upper boundary of the crosswalk's bounding box, *d*max is the maximum value of all d_{i}, and *N* is the number of traffic lights detected in the photo.

$$A_i = \frac{A_i^{'}}{A_{\max}} \qquad i \in (1, N)$$

where *A'* is the area of crosswalk's bounding box, and Ai is the maximum area of all A_i.

The distance is given a weight of *α*, while the area is given a weight of *β*, where *α*+*β*=1. The score *S* is formulated as

$$S = \alpha(1 - D) + \beta A$$

The traffic light with the highest score *S* is selected as the most relevant one. By maximizing this score, we effectively choose the most relevant traffic light for the given crosswalk.

Finally, after the second detection of crosswalk traffic light, the appropriate traffic light is selected, we crop it out from the image and save it for further processing. In our algorithm, the cropped image will be sent to the color detection module.

3.5 **Color detection**

To improve the color detection accuracy, we specifically designed a CNN for color recognition. This specifically designed CNN is used to detect the color of cropped image of the crosswalk traffic light. First, the input image is resized to 32×32 pixels and normalized. Second, the resized image is sent to the convolution and activation layers. The first convolution layer uses 32 filters with a kernel size of 3×3, a stride of 1, and padding of 1. The second convolution layer employs 64 filters, also with a 3×3 kernel, a stride of 1, and padding of 1. This is followed by a pooling layer with a kernel size of 2, a stride of 2, and no padding, typically used to reduce spatial dimensions and the number of parameters. Third, the multi-dimensional output is sent to the flattening and fully connected layers. A flattening step converts the multi-dimensional output to a one-dimensional array suitable for processing by fully connected layers. The first fully connected layer connects this flat output to 128 neurons, followed by a ReLU activation function that introduces nonlinearity and enhances the model's learning ability. The dropout layer, with a dropout probability ρ of 0.5, is then applied to help prevent overfitting by randomly omitting part of the neuron output. Finally, a second dense layer is used for binary classification, connecting 2 output neurons before the Softmax layer.

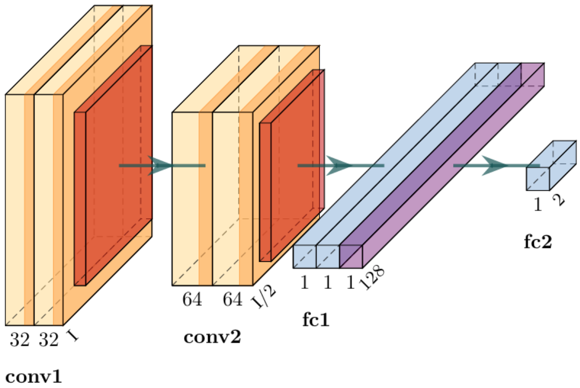


Figure 10. The structure of CNN

4. Experiments

4.1 Dataset collection and processing

To train and use our algorithm effectively, the dataset needs to be organized in a specific format. We made the labels on the website makesense.ai[12], an open-source annotation tool that enables drawing bounding boxes around objects in images to label them for our training tasks.

The images used are partly taken by ourselves and partly from ImVisible dataset on GitHub. For our specific goal of crosswalk traffic light and crosswalk detection, we selected 600 images from these two sources. These images include crosswalk traffic lights in various scenarios. In the labeling process, we annotated both the crosswalk traffic lights and the crosswalks respectively. For the crosswalk traffic lights, we use a single class label. Each traffic light in the image is manually annotated with a bounding box. For crosswalks, the labeling process involved more detailed considerations. Shown in Figure 11, we define the bounding box for crosswalks by selecting the upper edge of the crosswalk as one side of the rectangle and extending its downward to encompass the entire crosswalk area. This approach ensures that the bounding box accurately reflects the shape and orientation of the crosswalk, which can vary depending on the angle and perspective from which the image is taken.

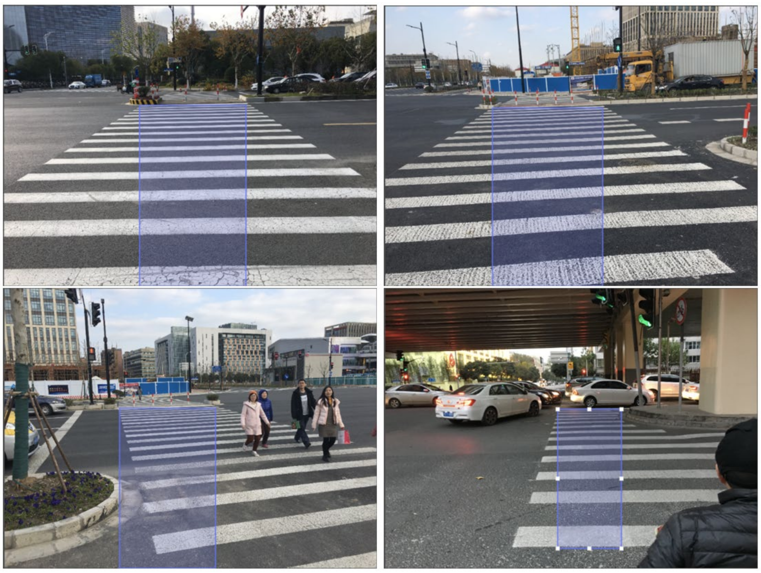


Figure 11. Crosswalk Labeling

After labeling, the images and their corresponding annotations are divided into three subsets: training, validation, and test sets. The allocation ratio of training, validation, and test sets is 7:2:1. This distribution ensures that the model is trained on a sufficient variety of data, while also allowing for robust evaluation and tuning during the validation and testing phases.

4.2 Analysis of dataset features

As shown in figure 12(a), the spatial distribution of traffic lights across the dataset is high. Most of the objects are located close to the center, but at the same time, there is a distribution of objects over the locations in such a sporadic way that it clearly belongs to different distances, perspectives, and angles. The size distribution of traffic light objects emphasizes that most of the objects are small in size, with a very small width and height in terms of their sizes in the image.

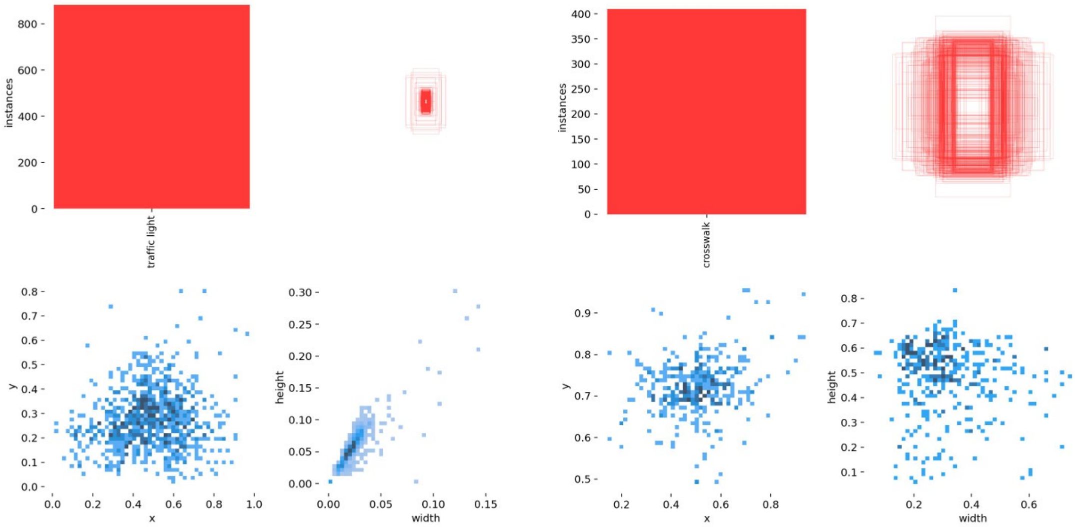


Figure 12. (a) Traffic Light Dataset Features (b) Crosswalk Dataset Feature

In figure 12(b), the *y* values of the bounding box center coordinates, concentrated be tween 0.7 and 0.8, indicate that most crosswalks are located in the lower portion of the images, leaving sufficient space above for selecting the right traffic lights we want. Additionally, the height values, primarily between 0.5 and 0.6, show that crosswalks are tall enough to effectively exclude traffic lights that are invalid in our detection.

4.3 Performance of traffic lights and crosswalk detection

Table 1 shows that the accuracy of the traffic lights detection model using SE module and the crosswalk detection model using CBAM module is significantly higher than that of the original YOLOv5s model. In addition, mAP@0.5 has a high value. However, the recall rate is slightly lower by comparison. This suggests that these models detect fewer overall targets, but the targets they detect are identified with very high accuracy. In contrast, the unmodified original model detected a higher number of targets but with lower accuracy. Since the goal is to help blind people cross traffic lights, it is vital to avoid false alarms. While YOLOv5 without SE has a higher recall rate, it can generate more false positives, which can be dangerous for visually impaired users. This highlights the advantages of an improved model.

Table 1. Performance metrics for traffic light and crosswalk detection

	Precision	Recall	mAP@0.5	mAP@0.5:0.95
Traffic Light YOLOv5s	0.911	0.954	0.967	0.611
Traffic Light YOLOv5s+SE	0.951	0.932	0.968	0.587
Crosswalk YOLOv5s	0.975	0.976	0.971	0.813
Crosswalk YOLOv5s+CBAM	1.000	0.975	0.975	0.779

As shown in figure 13, the loss of the model decreases very fast, especially in the initial training stage, both box loss and obj loss decrease rapidly, indicating that the model is rapidly converging. mAP@0.5 is significantly higher than mAP@0.5:0.95, indicating that the detection performance of the model is better when the threshold of IoU is set to 0.5. Therefore, in this case, setting the IoU threshold to 0.5 is a more appropriate choice, because it can bring higher detection accuracy.

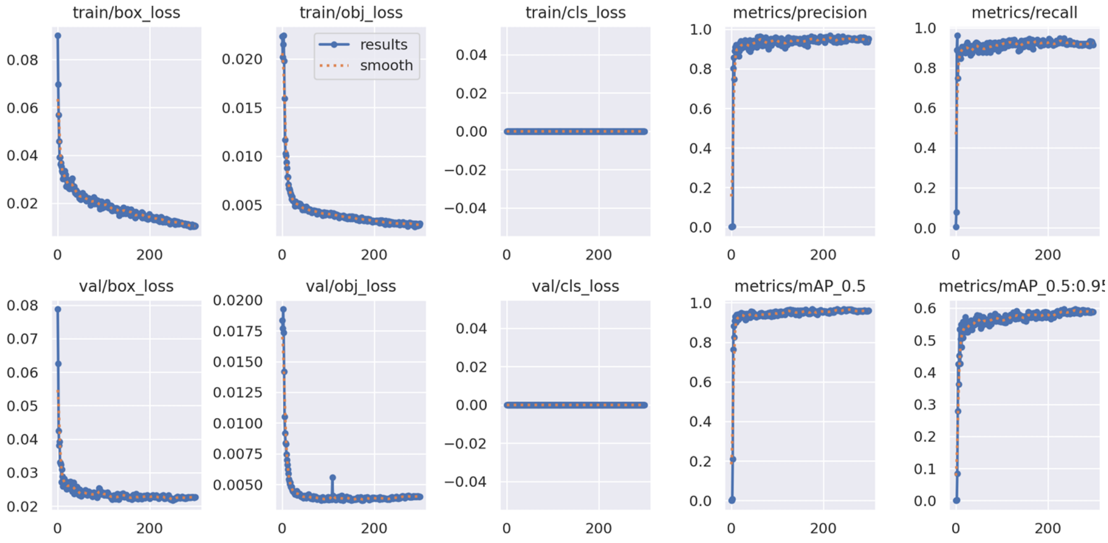


Figure 13. Training results of traffic light detection

It can be seen from figure 14 that loss gradually decreases during the training process for both box_loss and obj_loss, indicating that the model is continuously optimized. Also, train loss and val loss decrease gradually and converge to similar values. This shows that there is no obvious overfitting phenomenon in the training process of the model, and the performance of the model is consistent on the training set and the verification set. Both mAP@0.5 and mAP@0.5:0.95 show good performance, especially the high value of mAP@0.5:0.95, indicating that the model is stable under different IoU thresholds and has strong generalization ability. This means that the model not only has good detection performance at looser IoU thresholds (such as 0.5), but also at more stringent thresholds (such as 0.95).

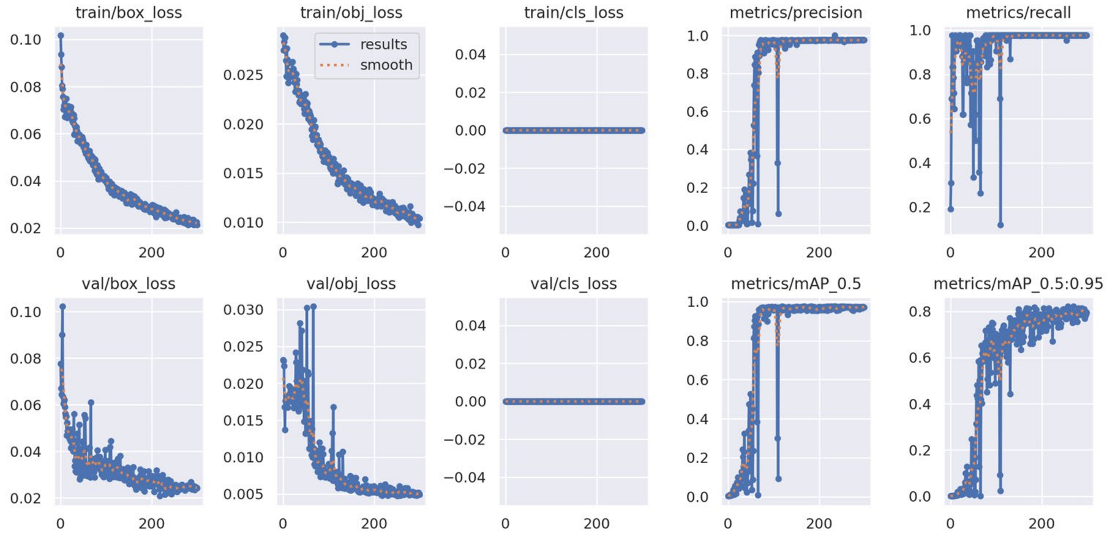


Figure 14. Training results of crosswalk detection

For figure 15(a), the model success fully identifies all the crosswalks and accurately frames the upper and lower boundaries of the crosswalks. This precise boundary framing not only ensures the accurate detection of the crosswalks, but also provides a crucial help for the subsequent auxiliary traffic light selection algorithm to ensure that the algorithm can be analyzed in the right area. Each detection was accompanied by a fairly high confidence level (0.9), demonstrating a high degree of trust in the identification results by the model. For image 15(b), except for the traffic light in one picture that is too small to be recognized, all the other small target traffic lights are effectively and accurately detected, and the detection results have high confidence (0.7-1.0). This shows that the model can provide reliable and accurate identification results even when dealing with small size targets.

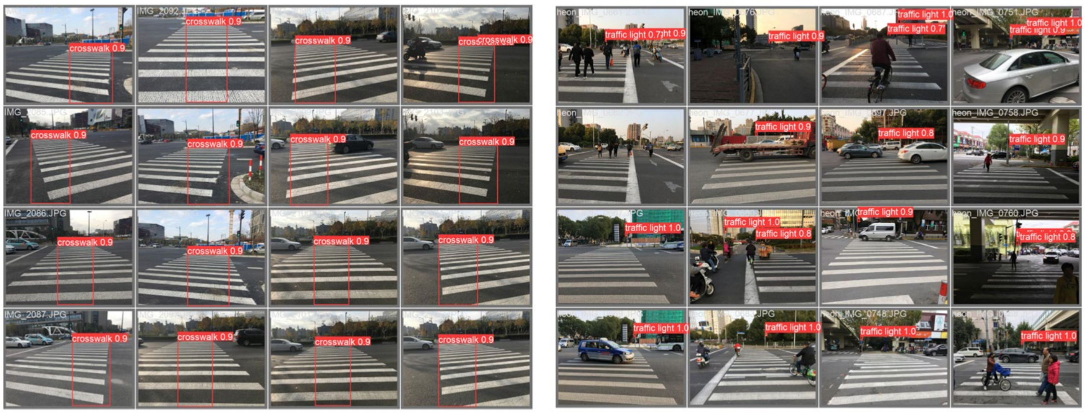
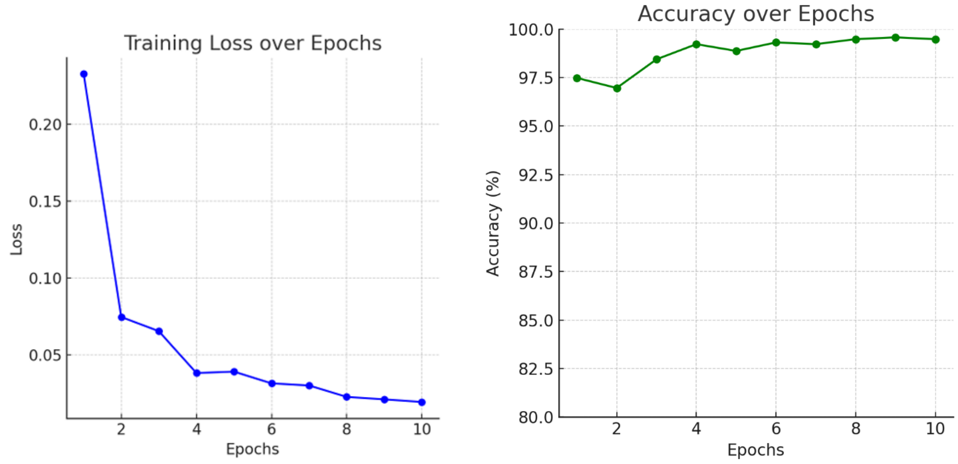


Figure 15. (a)Traffic Light Dataset Features (b)Crosswalk Dataset Features

4.4 Performance of color detection

We use CNN to implement color detection. Figure 16 represents both training loss and accuracy with 10 epochs. The loss, shown in figure 16(a), decreases from approximately 0.23 to about 0.02, while the accuracy in the figure 16(b) increases to a peak of about 99.57%, with only a slight fluctuations between epochs.



(a) (b)

Figure 16. CNN train results

We perform a sensitivity analysis to evaluate the robustness of our traffic light classification model under different image conditions. Specifically, we explore how changes in brightness, contrast and noise affect the performance of the model. We set a variable *x* to the six values with common difference respectively (as shown in table 2) and change the dataset with modified brightness, contrast and noise based on the value of *x* separately. For brightness variation, we set

$$I_{new} = I_{orig} \times \text{factor}_{\text{brightness}}$$

where $\text{factor}_{\text{brightness}}$ is randomly chosen from $(1-x,\ 1+x)$, and for contrast variation, we set

$$I_{new} = (I_{orig} - 128) \times \text{factor}_{\text{contrast}} + 128$$

where $\text{factor}_{\text{contrast}}$ is randomly chosen from $(1-x,\ 1+x)$. Finally, for noise variation, we set

$$I_{new} = I_{orig} + \text{noise}$$

where noise ~ $\mathcal{N}(0, \sigma)$, $\sigma = \text{factor}_{\text{noise}} \times 255$, $\mathcal{N}(0, \sigma)$ is a normal distribution with a mean of 0 and a variance of σ.

As presented in table 2, the sensitivity analysis shows that contrast is the most robust factor for CNN networks to distinguish between red and green lights at all contrast levels, and the model maintains very high accuracy, consistently above 98%, reaching a maximum of 99.65% with a contrast coefficient of 1.3. Brightness is the second robust factor, with accuracy consistently higher than 92%, reaching a peak of 99.57% at brightness coefficients of 0.9 and 1.1. While the network performs well at different brightness levels, it is slightly more affected by variation of brightness than contrast. Noise has a greater impact on performance. As the noise factor decreases, the accuracy increases from 54.73% at 0.6 to 99.48% at 0.1.

Table 2. Sensitivity analysis

Brightness Factor	Brightness Accuracy (%)	Noise Factor	Noise Accuracy (%)	Contrast Factor	Contrast Accuracy (%)
0.5	92.71	0.1	99.48	0.5	98.27
0.7	99.13	0.2	98.96	0.7	99.13
0.9	99.57	0.3	90.55	0.9	99.57
1.1	99.48	0.4	77.36	1.1	99.57
1.3	99.22	0.5	65.13	1.3	99.65
1.5	99.05	0.6	54.73	1.5	99.57

4.5 System performance evaluation

The traditional YOLOv5 model detects and labels both vehicle traffic lights and crosswalk traffic lights in the image without making any distinctions, which results in difficulties in selecting the correct crosswalk traffic lights. Therefore, for comparison, we select the traffic light with the largest pixel size from multiple traffic lights detected by traditional YOLOv5. We also change some parameters in our model to show how the accuracy changes.

Table 3 compares the different models based on a few key parameters. Score weight is a factor to adjust the weights between the object's distance to the crosswalk and in terms of area. Cropping(*m*, *n*) is the percentage of the image cropped at the two sides of the detected object before classification according to the coordinate of the crosswalk's bounding box. Color detection is done through either bounding box classification from YOLOv5 or a separate CNN for more precise color detection. The last column, accuracy, gives the overall performance of each configuration.

Table 3. Comparison of models with different strategies

Model Type	Score Weight (α, β)	Cropping (m, n)	Color Detection Method	Accuracy
1. YOLOv5+Pixel	N/A	Full Image	YOLOv5 Self-Classifies	0.920
2. Our Algorithm	0.5, 0.5	30, 20	CNN after Cropping	97.2%
3. Our Algorithm	0.0, 1.0	50, 20	CNN after Cropping	97.2%
4. Our Algorithm	1.0, 0.0	50, 20	CNN after Cropping	95.3%
5. Our Algorithm	0.5, 0.5	Full Image	CNN after Cropping	96.7%
6. Our Algorithm	0.5, 0.5	50, 20	CNN after Cropping	97.5%

Simulation results show that our algorithm achieves higher performance. Comparing the results obtained from 1 and 3, which are full image cases, YOLOv5 combined with maximum pixel detection achieves a relatively lower accuracy of 92.0%. However, our algorithm accuracy improves significantly to 96.7%. This shows that our algorithm can substantially enhance the model's overall accuracy for detecting small object and specific color. Comparing the results obtained from model type 2, 5 and 6, which have the same score weight, we can see that the lowest accuracy 95.3% is obtained with full image. These results show that after determining the crosswalk location, restricting the area for the second detection can eliminate interference from other traffic lights, thereby improving detection efficiency. Comparing the results obtained from model type 3, 4 and 6, which have the same cropping parameters, the accuracy 97.5% with score weight (0.5, 0.5) is the highest. These results show that in situations with multiple lights, focusing solely on one aspect will decrease the accuracy, and the distance to the crosswalk and the size of the traffic lights are also important.

Table 4 highlights the superior performance of our proposed algorithm compared to the traditional algorithm. Our algorithm shows a notable increase in precision (98.3% vs. 92.8%) and accuracy (97.5% vs. 92.0%), indicating better overall detection reliability. Additionally, it reduces the number of incorrect detections significantly (10 vs. 43) while maintaining a high recall rate

(99.2%).

Table 4. Performance comparison of our algorithm and traditional YOLOv5+Pixel

Model	Correct	Not Detected	Incorrect	Precision	Recall	Accuracy
Our Algorithm	585	5	10	98.3%	99.2%	97.5%
YOLOv5+Pixel	552	5	43	92.8%	99.1%	92.0%

5. Conclusion

The paper presents a novel crosswalk traffic light detection algorithm designed to resolve the specific challenges faced by the visually impaired when crossing crosswalks. The proposed algorithm effectively resolves multi-light interference by associating traffic lights with the corresponding crosswalk, significantly improving detection accuracy. SE and CBAM are used to enhance the detection of small objects and crosswalks under varying conditions, respectively. Experimental results demonstrate that our algorithm performs better than traditional YOLOv5 algorithm combined with maximum pixel detection, especially in distinguishing crosswalk traffic lights from other traffic lights, which effectively improves the travel safety of the visually impaired.

Accurate detection is crucial for safety. Although our algorithm has achieved better results, it is still not 100% safe. In the future, we need to further analyze the images with detection errors to improve our algorithm and expand the dataset to include more diverse crosswalk and traffic light configurations.

Reference:

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84-90, 2017.

[2] R. Girshick, J. Donahue, T. Darrell, et al., "Region-Based Convolutional Networks for Accurate Object Detection and Segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 1, pp. 142-158, 2015.

[3] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016.

[4] T.-Y. Lin, P. Goyal, et al., "Focal Loss for Dense Object Detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 2, pp. 318-327, 2020.

[5] G. Jocher, "Yolov5," GitHub, 2020. Available: https://github.com/ultralytics/yolov5

[6] P. S. Swami and P. Futane, "Traffic Light Detection System for Low Vision or Visually Impaired Person Through Voice," in *2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA)*, Pune, India, 2018, pp. 1-5.

[7] S. Eto, Y. Wada, and C. Wada, "Convolutional Neural Network Based Zebra Crossing and Pedestrian Traffic Light Recognition," *Journal of Mechanical and Electrical Intelligent System*, vol. 6, no. 3, pp. 1-11, 2023.

[8] V. Rao and H. Nguyen, "A computer vision based system to make street crossings safer for the visually impaired," *Journal of High School Science*, vol. 8, no. 2, pp. 253-266, 2024.

[9] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, June 2018.

[10] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "CBAM: Convolutional block attention module," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sept. 2018.

[11] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proc. of the 27th Int. Conf. on Machine Learning (ICML-10)*, 2010, pp. 807-814.

[12] https://www.makesense.ai/

致谢

与这篇论文相关的故事最早可以追溯到 2022 年 11 月，我们在珠江路上偶遇了一位盲人老爷爷。繁华的珠江路，车来人往，老爷爷拄着盲杖，但是由于盲道被电瓶车占用，老爷爷走得艰难而危险，让我们很难过。从那时候起，我们就开始调研南京盲人出行的现状。

本文首先要感谢的是调研过程中给我们提供帮助的老师和朋友，特别是一些视觉障碍朋友。论文工作的初步想法源于去盲校的一次交流。半年前，在南京市盲人学校老师的带领下，我们得以和该校多名学生进行交流。交流过程之中，一位同学说他们出门时过马路非常不方便。他不是盲人，是视弱者，他们无法清晰看到马路对面的红绿灯，安全地通过斑马线。在如今这个汽车都有了智能红绿灯信号识别系统的时代，他们却依旧面临着如何安全通过斑马线的难题。在这一年多和盲人朋友的接触中，我们了解到与我们平时想的不同，实际上通过语音辅助，他们大部分能使用智能手机，而且用得还不错，照相这些功能他们基本都能使用，甚至还会拍照发朋友圈。于是，我们决定设计结合手机拍照，帮助视弱群体安全过马路的交通灯识别算法。我们一开始尝试用已有的机器学习算法来解决这个问题。然而，在实际操作中我们发现该方案并不能满足我们排除干扰红绿灯，在多个红绿灯中选择出正确的人行横道红绿灯的要求。通过对图片的分析，我们发现人行横道红绿灯总是与人行横道共同存在，于是我们利用人行横道作为我们判别正确红绿灯的参照，最终达到了更高的检测率。在这一过程中，同学、朋友、还有在各地的叔叔阿姨们为我们拍摄了来自不同城市的人行横道红绿灯图片，极大地丰富了数据的多样性，对此，我们深表感谢。

我们也衷心地感谢东南大学张川教授为我们项目所提供的一切帮助，张老师同时也是张景涵同学英才计划的导师。从发现问题到完成算法，我们中途遇到过众多难题。无论是大到方案设计，还是小到程序上的报错，张老师总是无偿地给予我们引导性的指导。相较于直接告诉答案，张老师教给我们的是解决问题的方法和严谨的学术思维，让我们受益匪浅。同时，张老师开放服务器给我们使用，大大的缩短了我们程序运行的时间，显著提升了我们的效率。

我们还要感谢互联网提供的各种资源，感谢 CSDN 社区、GitHub 上大家的无私分享，作为受益者，我们也将代码上传，希望有兴趣的同学下载交流。

作为队友，我们配合默契，分工明确。张景涵同学主要负责方案设计，测试，田桢干同学负责编程和测试，我们共同拍摄、收集和处理了实验所需要的图片，并撰写了论文，论文第一节、第二节和第三节部分和结论主要由张景涵同学撰写，论文第三节部分和第四节主要由田桢干同学撰写。在此期间，要感谢我们彼此一直以来的理解和配合，特别是冲刺阶段互相的鼓励和奋战，令人难忘。

感谢南京外国语学校给了我们一个宽松的学习环境，让我们能够自由探索，让我们能结识更多优秀的老师和同学。

最后，感谢丘成桐中学科学奖给了我们这个成长的机会。作为一名高中生，我们得以在上大学之前就较为系统地了解学科知识，实验方法，论文写作。这不仅仅帮助我们掌握了知识与技术，也坚定了我们在这条路上继续钻研下去的决心。