

Article

Real-Time Pattern-Recognition of GPR Images with YOLO v3 Implemented by Tensorflow

Yuanhong Li ^{1,2,3}, Zuoxi Zhao ^{1,2,*}, Yangfan Luo ^{1,2} and Zhi Qiu ^{1,2}

¹ College of Engineering, South China Agricultural University, Guangzhou 510642, China;

liyuanhong@stu.scau.edu.cn (Y.L.); luoyangfan@stu.scau.edu.cn (Y.L.); qizhi@stu.scau.edu.cn (Z.Q.)

² Ministry of Education Key Technologies and Equipment Laboratory of Agricultural Machinery and Equipment in South China, South China Agricultural University, Guangzhou 510642, China

³ Department of Biological and Agricultural Engineering, Texas A&M University, College Station, TX 77843, USA

* Correspondence: xxcs@stu.scau.edu.cn

Received: 16 September 2020; Accepted: 10 November 2020; Published: 12 November 2020

Abstract: Artificial intelligence (AI) is widely used in pattern recognition and positioning. In most of the geological exploration applications, it needs to locate and identify underground objects according to electromagnetic wave characteristics from the ground-penetrating radar (GPR) images. Currently, a few robust AI approach can detect targets by real-time with high precision or automation for GPR images recognition. This paper proposes an approach that can be used to identify parabolic targets with different sizes and underground soil or concrete structure voids based on you only look once (YOLO) v3. With the TensorFlow 1.13.0 developed by Google, we construct YOLO v3 neural network to realize real-time pattern recognition of GPR images. We propose the specific coding method for the GPR image samples in Yolo V3 to improve the prediction accuracy of bounding boxes. At the same time, K-means algorithm is also applied to select anchor boxes to improve the accuracy of positioning hyperbolic vertex. For some instances electromagnetic-vacillated signals may occur, which refers to multiple parabolic electromagnetic waves formed by strong conductive objects among soils or overlapping waveforms. This paper deals with the vacillating signal similarity intersection over union (IoU) (V-IoU) methods. Experimental result shows that the V-IoU combined with non-maximum suppression (NMS) can accurately frame targets in GPR image and reduce the misidentified boxes as well. Compared with the single shot multi-box detector (SSD), YOLO v2, and Faster-RCNN, the V-IoU YOLO v3 shows its superior performance even when implemented by CPU. It can meet the real-time output requirements by an average 12 fps detected speed. In summary, this paper proposes a simple and high-precision real-time pattern recognition method for GPR imagery, and promoted the application of artificial intelligence or deep learning in the field of the geophysical science.

Keywords: pattern-recognition; real-time; YOLO v3; TensorFlow; electromagnetic wave

1. Introduction

In the application of ground-penetrating radar (GPR) engineering detection, the following three cases are the most common: (1) Inspection of the atypical situation of reinforced concrete structures such as bridges, tunnels, or public roads, or the number of steel bars inside those structures; (2) locating certain objects underground, such as archaeological research; (3) evaluating and measuring the distribution of hollows, voids, or soil firmness in highways, bridges, and tunnels. Nonetheless, the outcomes, after GPR detection, are often judged by the worker's experience to recognize the location and size information of the target [1,2]. Actually, these kinds of evaluations using GPR image are not infeasible, but consume a lot of manpower and material resources. For example, the 3D radar

launched currently by MALA can collect data on multiple channels. Suppose that a 3D GPR with 22 acquisition antennas can generate 22 GPR images at the same time; if we evaluate its image outcome by traditional method, the analysis work will be very inefficient [3,4]. Besides, with the continuous development of 3D radar imaging technology, especially for multi-channel GPR, efficient and intelligent AI algorithms can not only output the analyzed results automatically, but also can fulfill the application demands of the underground exploration engineering, such as long-distance detection of reinforced concrete of roads; investigation of large-area cavities of bridges and tunnels; early warning of urban road collapse.

In terms of using artificial intelligence to identify GPR imagery, Sonoda and Kimoto (2018) adapted the finite-difference time-domain (FDTD) to simulate multiple GPR images, and trained a 9-layer deep neural network (DNN) model to extract feature maps that contain many hyperbolic signals of underground objects [5]. Finally, they obtained the characteristics of electromagnetic wave intensity from the curve signal and identified six materials with 80% accuracy. Because of the limited quantity of DNN layers, this method lacked accuracy of the identified materials and is limited to its sample selection. Aydin and Yüksel (2017) adapted the GprMax simulation API program to generate GPR B-scan images, then proposed to combine two convolutional layers and pooling layers to classify the electromagnetic wave, but they did not involve in-depth or improved research in classified speed [6]. Dinh et al., (2018) validated performance with traditional GPR images processing algorithms and convolutional neural network (CNN). At last, the reinforcement in GPR images were positioned and inspected automatically. After analyzing 26 bridge decks GPR data, they achieved a recognition accuracy of $99.60\% \pm 0.85\%$; but its detecting speed did not fulfill the engineering demands of real-time outcome [7]. Pham and Lefèvre (2018) used the faster-RCNN framework to detect hyperbola reflections from many B-Scans generated from gprMax toolbox and the results show that faster-RCNN framework can provide significant improvements to deal with GPR data [8–11]. Kechagias-Stamati et al. proposed a CMNet network for synthetic aperture radar (SAR) image target recognition based on convolutional neural network. The network adds center loss and softmax training process to the feature layer of SAR images [12]. In order to improve the target recognition rate of SAR image, both intra class aggregation and inter class separation were considered. However, this significantly reduces the utilization of hyperbolic features in GPR images. Dou et al. (2017) proposed a novel technique called column-connection clustering (C3) algorithm to separate hyperbolae in GPR images, and obtain hyperbolic signatures. This method can also be used for real-time detection [13]. The fitting speed is 0.73 s per hyperbola. However, the number of hyperbolic objects in GPR images is often large. Compared with the Yolo V3 recognition method proposed in this paper, the recognition speed of 12 frames per second per image is more dominant. In addition, they only test the scattered hyperbola. On this basis, this paper also tests the hyperbola-intensive samples, and achieves ideal detection effect. Pham et al. (2020) proposed an improved YOLO structure called YOLO-fine to detect very small objects from aerial and satellite remote sensing images. However, for GPR images, we not only need to detect small objects, but also need to identify intensive hyperbolic features [14]. Obviously, in many engineering cases, very small hyperbolic features are not the common situations.

2. Materials and Methods

2.1. YOLO v3 Feature Extractor

YOLO v3 is a classical pattern-recognition algorithm based on darknet-53 CNN architecture proposed by Joseph Redmon in 2018 [15]. It is currently a marketable object detection algorithm. Most importantly, it has ultra-fast detected speed than SSD, but almost as accurate as faster-rcnn [16,17]. The YOLO v3 basic framework contains the convolutional layer, batch normalization (BN) layer, and leaky rectified linear unit (ReLU) layer [18,19]. First, it is assumed that all input images are resized into 416×416 and three types of feature maps; then they go through a $32 \times 3 \times 3$ (filter numbers are 32 with 3×3 sizes) and $64 \times 3 \times 3$ convolutional layer, and output a 208×208 feature map with 64 channels. Where the second block network of YOLO v3 carried one residual block which includes zero padding, convolution and residual unit, and $128 \times 3 \times 3$ convolutional layer; it outputs some 104

$\times 104$ feature maps with 128 channels. The third YOLO v3 network contains two residual block, then go through $256 \times 3 \times 3$ convolutional layer, where the 4th block contains many residual shortcut to all 256×256 feature maps. In addition, this block makes vector concatenated operation of residual shortcut to reduce the gradient explosion and outputs 52×52 feature maps with 384 channels. With up-sample, some 52×52 feature maps are outputted for YOLO v3 to detect small-scale objects [20]. Similarly, the fifth block outputs many 26×26 feature maps for detecting medium-scale targets. At last, network still passes by many residual shortcut connection blocks which include zero padding, convolution, and residual unit. Finally, YOLO v3 designed a $255 \times 1 \times 1$ convolution layer to output 13×13 feature maps with 255 channels for detecting big objects [21]. In general, YOLO v3 can detect images on three different scales with 32×32 , 16×16 , and 8×8 feature maps, where the first detected operation layer is at 82th layer; its stride takes 32 to generate 13×13 feature maps. The second up-sampling operation is at 94th and the third detection layer is at 106th layer, which produces a feature map with dimensions $52 \times 52 \times 255$. Overall architecture of YOLO v3 is shown in Figure 1. In addition, we used the K-means clustering to select bounding box priors in YOLO v3.

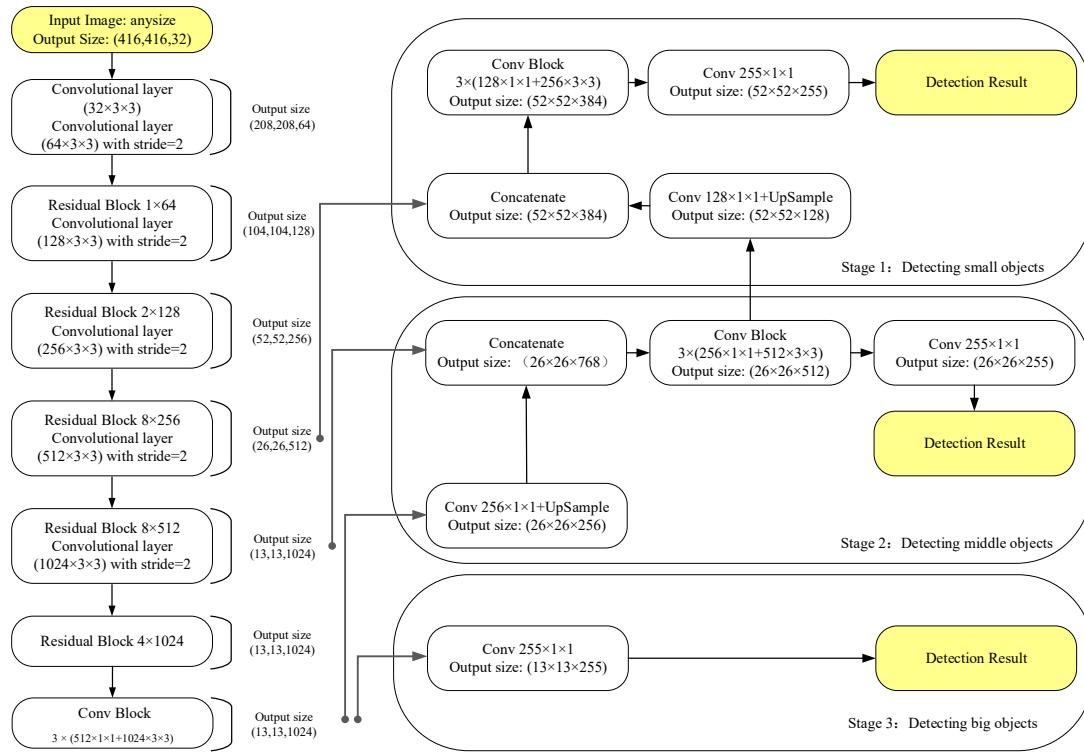


Figure 1. You only look once (YOLO) v3 architecture.

The following Figure 2 shows a part graph of YOLO v3 exported from TensorBoard of TensorFlow visualization API, which was actually a neural network connection diagram for YOLO's second up sampling. The TensorBoard can show the output and input tensor variables at each node, in addition, it can show the dependency between the tensor operations through some edges. The conv2d here is abbreviated for the convolution layer or block in Figure 1. Similarly, Leaky relu is denoted as ReLu layers in Figure 1 and batch normalization is denoted as BN layer. We can visualize all concatenation operation of each stage of YOLO v3 through TensorBoard, such as the attributes behind the convolution layer indicate that the input and output tensors correspond to this convolutional layer. The loss represents the value of the current convolutional layer after passing through the optimizer.

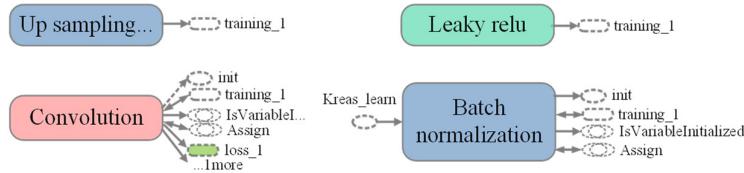


Figure 2. YOLO v3 visual format in TensorBoard.

2.2. Bounding Box Encoding Strategy

Soil objects underground are regularly sensitive to electromagnetic waves caused by its physical properties [22,23]. Most of them appear as parabolic with openings downward or obvious energy reflection in electromagnetic waves format. GPR moves along the survey line and continuously collects a series of trajectories (A-scan) to form electromagnetic wave B-scan images [24]. Before YOLO v3 training, GPR images were collected in this way. As mentioned in 2.1 section, YOLO v3 outputs feature maps or cells by three different stages, and each bounding box is responsible for multiple categories [25]. Suppose that the input GPR images size still is 416×416 , as shown in Figure 3a, then the original picture can be divided into 13×13 cells. Those cells that are parabolic vertex M in the GPR images are responsible for predicting corresponding targets. When annotating GPR image samples, we make the center of ground truth box (rectangle A1B1C1D1) to correspond to the position of parabola apex. Red box in Figure 3a contains the midpoint of target wave; rectangle A1B1C1D1 marked as red solid line is ground truth bounding box and ABCD marked as red dotted line represents the predicted box. Figure 3b shows the encode ways of ground truth box in YOLO v3. Point A₁ is the top left corner of box; t_x and t_y are the pixel position of point A₁ in GPR image. Zx and Zy are noted as pixel width and height of each cell respectively. The width of B₁D₁ is marked as t_w and the height of C₁D₁ is marked as t_h. As shown in Figure 3c, each bounding box was attributed to one object score or confidence $P_i \in \{0,1\}$, 4 box coordinates (t_x, t_y, t_w, t_h), and one class score S_i. Here, S_i follows similarly as the one hot encoding method and S_i $\in [0,1]$. If S_i equals to 0, there was no current detected target in the GPR image; otherwise, if S_i equals to 1, it indicates that there exists current detected target. Finally, the feature map corresponds to 13×13 cells, and the output bounding box encoded tensor shape is $(13 \times 13, 6)$. If there were n GPR image samples, then all bounding box sizes corresponded to tensor $(n, 13 \times 13, 6)$. It is worth noting that if it is a non-hyperbolic target, such as the voids detected below, we will use the original encoding method of YOLO v3.

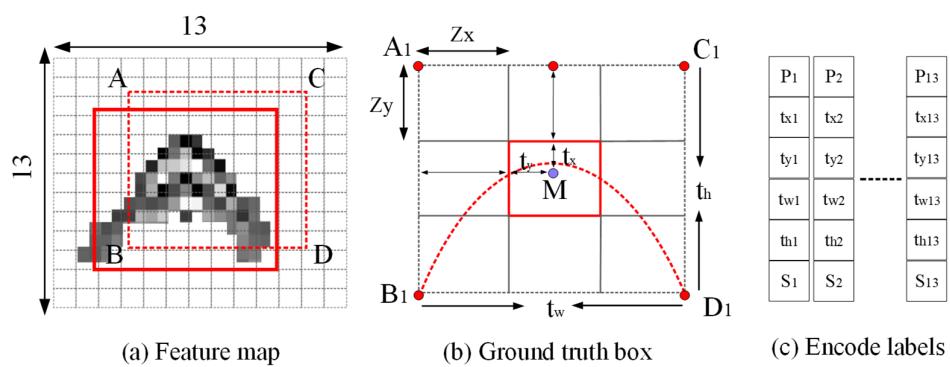


Figure 3. Improved YOLO v3 bounding box encoding: (a) Feature map of GPR target in YOLO v3; (b) Ground truth box of hyperbolic signal; (c) Encoding format of detecting GPR target.

2.3. Anchor Box Selection by K-Means Clustering

The K-means is an iterative algorithm that can divide data into K predefined clustering and cluster each point into specific data groups [26]. When YOLO v3 trains GPR sample data, anchor box can control skillfully the over fit recognition results of soil targets, because in the high-frequency electromagnetic wave reflection signal, when two target positions are relatively close, those close

parabolic vertices will be easily assigned to the same bounding box. K-means defines the size of bounding box through cluster analysis. Absolutely, K-means tries to keep clusters as different as possible at this point in order to minimize the sum of squared distances between all centers of data clusters [27,28]. First, we define K value and initialize the centroids by shuffling, then keeping iterating until there is no change in the centroids outcome. This is called expectation maximization [29]. Assuming that there are m samples, here we introduce a multi-sample function about K value:

$$F = \sum_{i=1}^m \sum_{k=1}^k \sigma_{ik} \|t(x, y) - \mu_k\|^2 \quad (1)$$

If the point $t(x, y)$ belongs to the K cluster, then $\sigma_{ik} = 1$ otherwise $\sigma_{ik} = 0$; at this time, μ_k can be considered as the centroid of $t(x, y)$. If the derivative of F function can minimize the equation solution, then the problem can be solved using the following formula:

$$\frac{\partial F}{\partial \sigma_{ik}} = \sum_{i=1}^m \sum_{k=1}^k \|t(x, y) - \mu_k\|^2 \quad (2)$$

Here, it is needed to distinguish F solution and recalculate the centroid after the last clustering iteration. Obviously, data points $t(x, y)$ are assigned to close clusters. Finally, we can recalculate each cluster centroid according to the following Equation (3) to reflect the situation of the new point allocation.

$$\frac{\partial F}{\partial \mu_k} = 2 \sum_{i=1}^m \sigma_{ik} (t(x, y) - \mu_k) = 0 \quad (3)$$

$$\mu_k = \frac{\sum_{i=1}^m \sigma_{ik} t(x, y)}{\sum_{i=1}^m \sigma_{ik}} \quad (4)$$

K-means uses data distance as the evaluated criterion to determine the selection of anchor box. Algorithm iteration is initialized at the beginning. In order to avoid the F function staying at the local optimal rather than global optimal, this paper adopts a variety of centroid initialization to run the K-means algorithm [30,31]. After filtering by K-means, the encode label of YOLO v3 for GPR image increases the data dimension. As shown in Figure 4, Y_{label} represents the encoded bounding box without increasing the dimension. This is the transpose of data matrix in Figure 3c. $Y_{K-Means}$ represents the encoded bounding box that have added n anchor boxes output by K-means clustering.

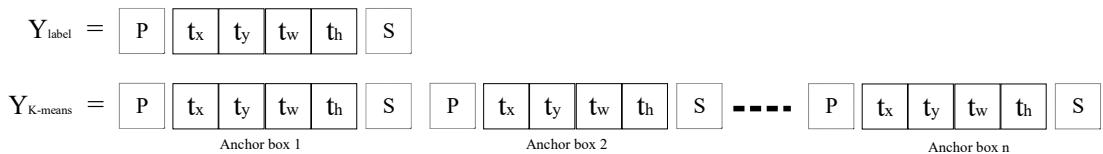


Figure 4. Encoding bounding box with multiple anchor box after k-means selection.

2.4. Principle Analysis of V-IoU Processing with NMS

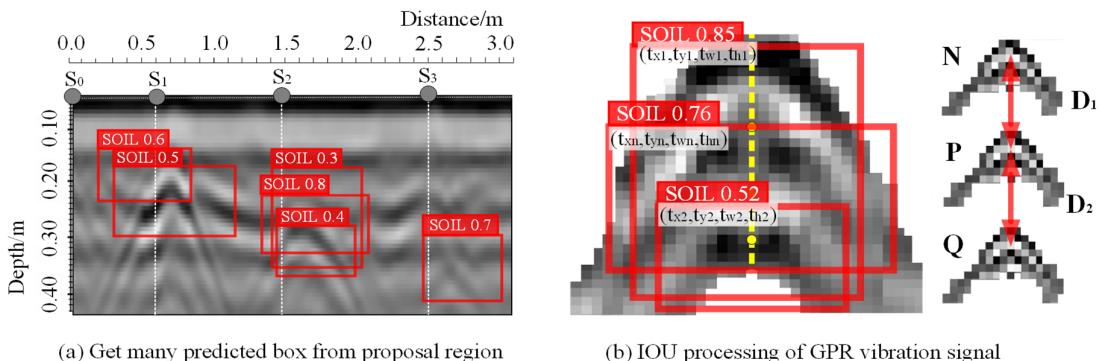
Non-maximum suppression (NMS) is commonly applied to extract the window with the highest score in detection algorithm, such as feature extraction in sliding windows, pedestrians in automatic driving, and vehicle recognition [32]. Similarly, in the GPR image, after feature maps are produced by the convolutional layer of YOLO v3 in the 3rd stage recognized by the classifier, for some underground targets, there are a large number of bounding boxes that cross each other or contain the same parabolic midpoint in one cell. The goal of NMS is to remove the detected redundant boxes and keep the best one. First, it is needed to mention here the intersection over union (IoU) score. IoU is a standard performance metric for image category segmentation problems [33]. For a given set from image, IoU defined by Equation (5) gives the ratio of intersection and union of the predicted bounding box and ground truth bounding box [34]. Suppose t represents the probability outputs of

pixel set N after filter by activation function in the GPR image; Y denotes the data set composed of ground truth bounding box; $Y \in \{0,1\}^M$ marks 0 for non-target pixels and 1 for target pixels.

$$\text{IoU} = \frac{I(t)}{U(t)} = \frac{\sum_{n \in N} t_n * Y_n}{\sum_{n \in N} (t_n + Y_n - t_n * Y_n)} \quad (5)$$

First, in YOLO v3, NMS can calculate the confidence C of the proposal region and sort the bounding boxes list. Second, NMS selects the predicted box with the largest score; then the IoU coefficients of other remain bounding boxes and the current box are calculated. If the IoU value is greater than the predefined threshold, NMS will delete this bounding box [35]. This is a complete iterative process in which NMS is applied to select the maximum score bounding box for one target. Then in the second iteration, the highest score box is still selected in the remaining boxes and those that exceed the predefined IoU threshold are deleted until all possible targets in the GPR image have been pick up.

After the YOLO v3 residual network and 1×1 convolutional layer, a large number of bounding boxes are generated on the region proposal area outputted by feature map. As shown in Figure 5a, S_0 denotes the starting position of GPR: S_1 , S_2 , and S_3 represent respectively the soil surface position of parabolic electromagnetic wave signal generated by three iron cylinders with buried depths of 0.25 m, 0.3 m, and 0.35 m, respectively. The soil dielectric constant is about 6.5 and the electrical conductivity is about 0.002 s/m. It can be seen that there are numerous prediction bounding boxes around each target. Now we focus on one parabola. In the process of YOLO v3 algorithm recognizing the target from GPR image, it is uncomplicated to misidentify the parabola originally belonging to one object as multiple targets because of the oscillating signal from electromagnetic wave [36]. The points N , P , and Q in Figure 5b represent three parabola generated by some strong conductive targets in depth direction of the soil. The number on a side of SOIL label represents the probability of being identified as a target, with the maximum value as 1 and the minimum as 0. YOLO v3 recognizes or locates those as three adjacent targets, but it is only one target, although their IoU threshold has been included in the predefined range. Therefore, this paper proposes the principle of V-IoU merging vacillate signals of similarity waves based on GPR images. Assume the location of ground truth box (red box) was marked as coordinate $(t_{xn}, t_{yn}, t_{wn}, t_{hn})$ and the locations of another two boxes which were marked by GPR echo signal oscillation were predicted as $(t_{x1}, t_{y1}, t_{w1}, t_{h1})$ and $(t_{x2}, t_{y2}, t_{w2}, t_{h2})$. Then the coordinate of ground truth box of point N can be denoted as $(t_{xn} + \frac{t_{wn}}{2}, t_{yn} - \frac{t_{hn}}{2})$. Similarly, the pixel coordinate of P is denoted as $(t_{x1} + \frac{t_{w1}}{2}, t_{y1} - \frac{t_{h1}}{2})$ and Q is denoted as $(t_{x2} + \frac{t_{w2}}{2}, t_{y2} - \frac{t_{h2}}{2})$. First, it is worth noting that we define a horizontal threshold β here and make the $(t_{xn} - t_{x1}) + (\frac{t_{wn}}{2} + \frac{t_{w1}}{2}) \in [-\beta, \beta]$. At the same time we define a longitudinal threshold α and made the $(t_{yn} - t_{y1}) - (\frac{t_{hn}}{2} - \frac{t_{h1}}{2}) \in [-\alpha, \alpha]$; if those parabolic midpoint or N , P , and Q points at the soil depth satisfy the horizontal and vertical critical values, we will liberate the limitation of IoU threshold and merge those prediction boxes. This is the core idea of V-IoU, for example, $D_i \in [-\alpha, \alpha]$ and $i \in R$.



(a) Get many predicted box from proposal region

(b) IOU processing of GPR vibration signal

Figure 5. V-IoU principle with non-maximum suppression (NMS): (a) The predicted boxes from proposal region; (b) The IoU processing of GPR vibration signal.

2.5. Loss Function and Learning Rate Adaptive Optimizer

Loss function of YOLO v3 in this paper is composed of mean variance and error [37]. Specifically, it is mainly divided into three parts for the calculation of offset losses, midpoint coordinate of parabola in GPR image prediction error gprErr, V-IoU prediction error viouErr, and classification error clsErr[38]. Here preset the weight of gprErr γ_{gpr} as 5 and the weight of viouErr γ_{viou} as 0.5 in order to rectify the domination of large target is weaker than the small target during detection. It can be expressed by the following formula:

$$\text{Loss} = \sum_{i=0}^{s^2} gprErr + viouErr + clsErr \quad (6)$$

After derivation, the loss function of this three parts can be expressed as:

$$\begin{aligned} gprErr = & \gamma_{gpr} \sum_{i=0}^{s^2} \sum_{j=0}^B I_{ij}^{tar} [(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2] \\ & + \gamma_{gpr} \sum_{i=0}^{s^2} \sum_{j=0}^B I_{ij}^{tar} [(\sqrt{w_i} - \sqrt{\hat{w}_i})^2 + (\sqrt{h_i} - \sqrt{\hat{h}_i})^2] \end{aligned} \quad (7)$$

where \hat{x}_i , \hat{y}_i , \hat{w}_i , and \hat{h}_i in the Equation (7) are denoted as predicted values by YOLO v3, x_i , y_i , w_i , and h_i expressed as training tag value; I_{ij}^{tar} indicates that if the object falls into the j-th position of lattice i-th bounding box, its value is either 1 or 0.

$$\text{viouErr} = \sum_{i=0}^{s^2} \sum_{j=0}^B I_{ij}^{tar} (C_i - \hat{C}_i)^2 + \gamma_{viou} \sum_{i=0}^{s^2} \sum_{j=0}^B I_{ij}^{notar} (C_i - \hat{C}_i)^2 \quad (8)$$

where \hat{C}_i in the Equation (8) is denoted as predicted value by YOLO v3, C_i expressed as training tag value, I_{ij}^{notar} indicates that the j-th bounding box of the object grid i does not contain the detection target.

$$\text{clsErr} = \sum_{i=0}^{s^2} I_i^{tar} \sum_{c \in \text{classes}} (P_i(c) - \hat{P}_i(c))^2 \quad (9)$$

where \hat{P}_i in the Equation (9) is denoted as predicted value, P_i is expressed as the training tag value. Figure 6 below shows a graph of the YOLO loss function node in TensorBoard; the input element were the loss output of conv2d_59, conv2d_67, and conv2d_75; where the input_1, input_2, and input_3 correspond to the gprErr, viouErr, and clsErr in Equation (6) respectively.

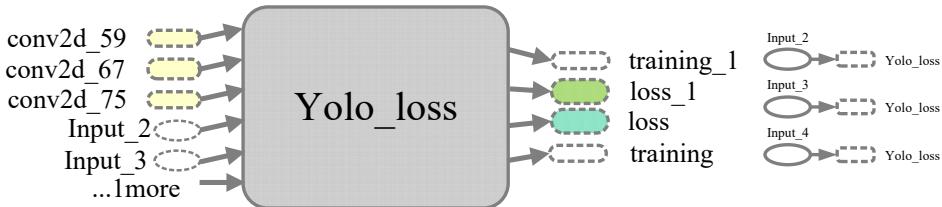


Figure 6. YOLO v3 loss visual format in TensorBoard.

When using the gradient descent method to optimize YOLO v3 loss value, even though the loss function have to be optimized near the minimum value, there still exists a large gradient. In this way, using a global learning rate will cause some serious problems, such as slow gradient convergence or unstable loss value. In order to solve this problem, this article uses the Adam algorithm which is a learning rate adaptive algorithm improved by the RMSProp algorithm proposed by Kingma in 2014 [39]. First, we set a default learning rate (0.001 in TensorFlow) and two exponential decay rates for moment estimation (default is 0.9 and 0.990 in TensorFlow); then initialize the moment variable and its time step count; finally, we continuously correct the deviation through biased moment estimation

to update the weight and learning rate. Figure 7 below shows two structural diagrams of Adam optimizers in TensorBoard.



Figure 7. Adam optimizer in TensorBoard.

3. Results and Discussion

3.1. Experimental Parameters

GPR model in this paper used the GX750-HDR (GEO AB Company, Sundbyberg, Sweden) of Swedish Guideline GEO AB Company. Sampling number collected for each channel was 412, sampling interval was 0.015 m, the coupling distance of GPR antenna preset was 0.14 m, and the diameter of the ranging wheel preset was 17 cm. GPR data preprocessing software was the REFLXW 7.5 which its copyright by K.J. Sandmeier. The training data set format adopted the COCO data format [40,41]. Here, we marked GPR image target for YOLO v3 training by the visual object tagging tool (VoTT) 2.1.0. Operating system was windows 10, and its processor model is Intel(R) Xeon (R) Gold 6130 CPU (Intel, Santa Clara, CA, USA) 2.10 GHz. Deep learning frameworks or related packages include the python 3.7, Keras 2.31, Tensorflow 1.13.1, cuDNN 7.4, Anaconda 3, Sklearn and GUDA 10.0. The main methods of preprocessing noise are: (1) Remove DC drift, (2) static correction cut, (3) gain, (4) remove direct ground wave, (5) remove high and low frequency signals, (6) horizontal smoothing. A total of 331 GPR image samples were collected in the experiment, of which the proportion of training set in whole data set is 70%, the validation set is 20%, and the test set is 10% in whole data [42]. In the YOLO v3 training stage, the batch size and subdivision of training sets are preset as 20. Epoch of each stage is preset as 51 and the learning rate is predefined as 0.001.

3.2. Anchor Boxes Selection by K-Means Clustering

After using VOTT tool to label all hyperbola targets from GPR images, there are 386 rectangular boxes containing parabola generated from the training dataset of ground truth images. Location parameters of ground truth box are composed of four corner coordinates of the rectangular box as (x_{min}, y_{max}) , (x_{min}, y_{min}) , (x_{max}, y_{max}) , and (x_{max}, y_{min}) . Obviously, we only need to take four parameters x_{min} , y_{min} , x_{max} , and y_{max} for clustering effect or silhouette coefficient analysis [43]. Silhouette coefficient is a significant evaluation index for clustering performance. Its value is commonly between $[-1, 1]$. When the silhouette coefficient is closer to 1, the cohesion and separation of K-means model are better. In Figure 8a, we adjusted the clustering or centroid number of K-means to 2; the maximum number of iterations is predefined as 200; after normalizing the x_{min} , y_{min} data, it can be seen that the clustering group of centroid were still relatively demonstrable. The silhouette coefficient output by the silhouette score function from sklearn module was 0.4839. Compared with Figure 8b, when the number of centroid was set to 3, there exist high-separation and low-cohesive phenomenon for the clustered groups after standardized data. Similarly, Figure 8c,d shows the clustering effect of x_{max} and y_{max} data when the clustering is set to 2 and 3. At this time, the silhouette coefficient output by the silhouette score function was 0.4868. After calculation, finally we got four anchor boxes values for training configuration parameters that consist of x_{min} , y_{min} , x_{max} , and y_{max} .

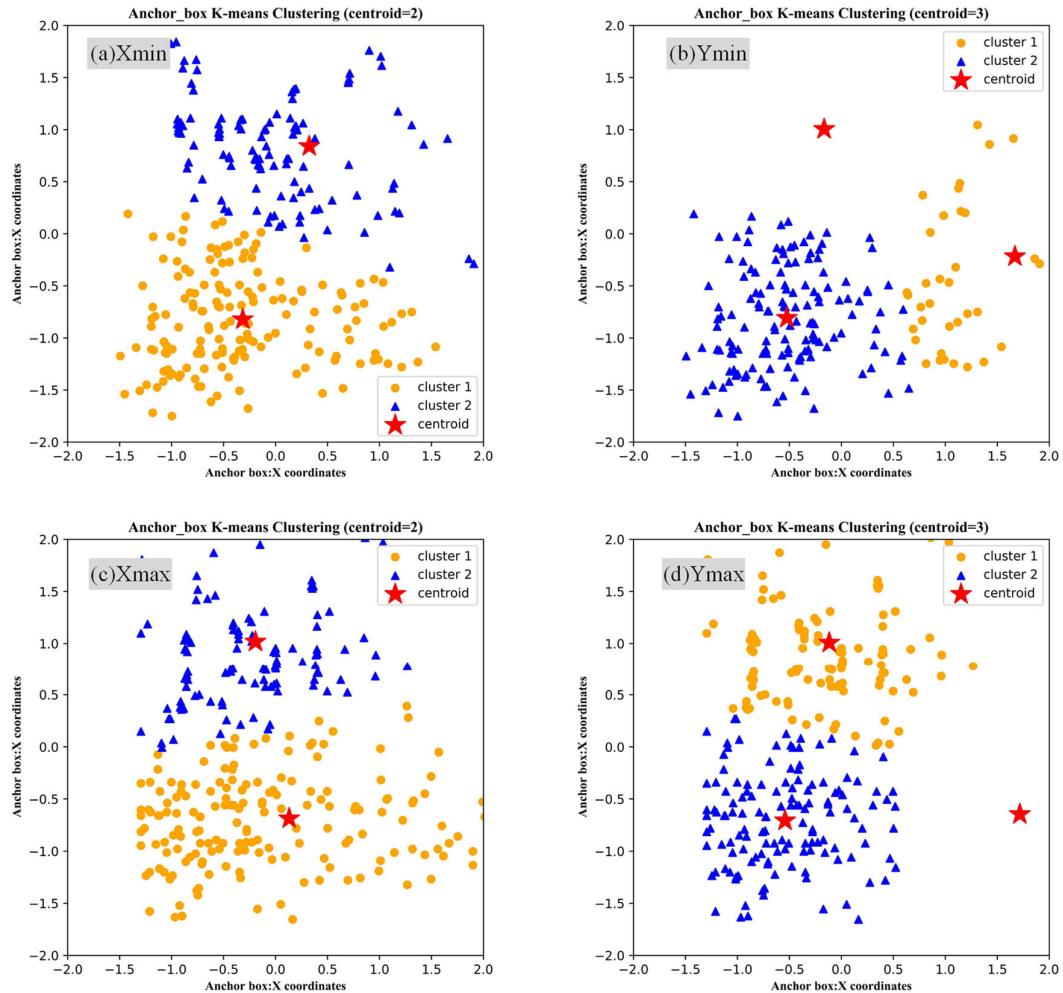


Figure 8. K-means clustering analysis of anchor boxes: (a,b) are related to Xmin and Ymin coordinates; (c,d) are related to Xmax and Ymax coordinates).

3.3. V-IoU and NMS Training Loss Performance

After derivation of 2.5 sections, IoU-YOLO v3 loss function contains three parts. The first part is the average error loss of the centroid position in GPR bounding boxes which is centroid position (t_x, t_y) relative to ground truth boxes. Here, the coordinate related to x axis of the predicted bounding box can be denoted as \widehat{b}_x which is equal to sigmoid (t_x) + C_x and its coordinates related to y axis can be denoted as \widehat{b}_y which is equal to sigmoid (t_y) + C_y . Obviously after weight processing, the smaller the loss value, the closer the centroid between the predicted coordinate ($\widehat{b}_x, \widehat{b}_y$) and the true value (b_x, b_y), the better the prediction performance of logical regression function. In the first training phase of YOLO V3 with the V-IoU and NMS, when the epoch was less than 10, the loss value began to decrease very fast. When in the second stage, the convergence speed of loss function became steady and slow. Comparing the blue curve without adding V-IoU in Figure 9, the training performance of YOLO loss function seemed equivalent in two stages, but the completion time of entire 83 epochs was 3 h and 57 min. This is because the local optimization produced by the training process will affect the algorithm calculation efficiency to update function weights by back propagation. For this reason, as can be seen from Figure 10, the loss value of IoU + NMS had been changing back and forward between 22.5 and 40, and three local optimal solutions that appear at the positions are indicated by five green arrows; however, V-IoU + NMS was relatively stable, and it is undemanding to perform global gradient descent to find the global optimal solution. In order to prevent data over fitting, the loss

function is considered to be sufficiently convergent; when the epoch was equal to 83 iteration was stopped.

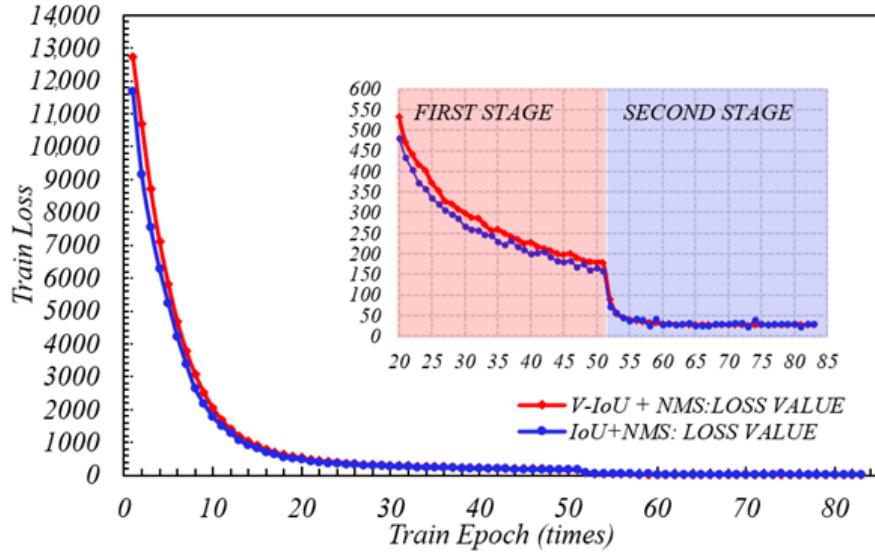


Figure 9. V-IoU versus IoU training loss (epoch: 0 to 83).

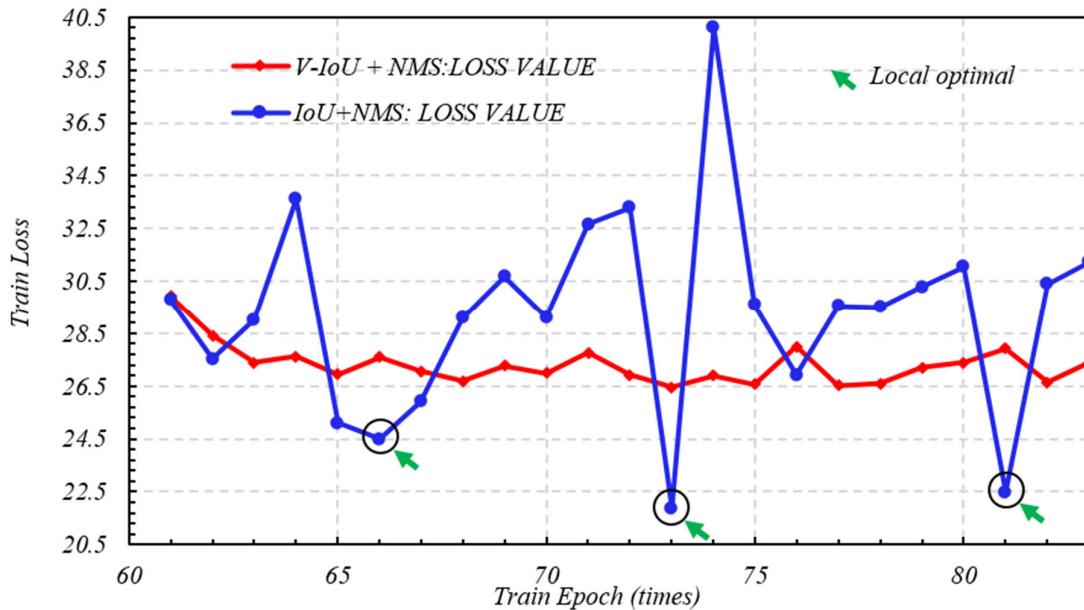


Figure 10. V-IoU versus IoU training loss (epoch: 60 to 83).

3.4. YOLO v3 Detection Effect

It can be known from the YOLO v3 network architecture in Section 2.1 that YOLO v3 can be detected on three feature maps of different scales and output after the input image size have been down sampled to 32, 16, and 8. Testing datasets contain three scenes for the real-time detected performance test, which cover the single class and multi-class pattern-recognition which include hyperbolic and voids features. The evaluation index refers the mean average precision (mAP) to training batches [44,45]. Assuming that P is denoted as the actual number of samples among target prediction, this is called precision. R is the recall rate, T is denoted as true positives, where $P =$

$\frac{TP}{(TP+FP)}$ and $R = \frac{TP}{(TP+FP)}$, where TP is the true positives and FP the false negatives; the mAP can be calculated by equation $\frac{\sum AP}{N_{classes}}$, where AP is denoted as the average precision. Figure 11 shows the improved detection effect of YOLO v3 with V-IoU on single class targets. The verified data set showed in Figures 11–13 were collected from the Soils research key Laboratory of South China Agricultural University. First, we detect the object's physical position through GPR, and then mark the hyperbola vertex by the marking button on the MALA GPR controller. Finally, we use the difference between the identified rectangle midpoint and the marker's value to determine the ground truth. Here, the V-IoU threshold was preset to 0.50. As can be seen from the figure, although some targets are small in the GPR image, the YOLO v3 detector can recognize it. This is because compared to YOLO v2, the V3 version has three detections, which are one down-sampled 13×13 and two up-sampled with 26×26 , 52×52 feature maps. In addition, YOLO v3 have added a series of convolutional layer with 3×3 or 1×1 size that increase appropriately the number of channels. Overall, in this situation, total 132 hyperbolas in GPR image were tested. The correct detection number is 121, missed targets number is 7, and false alarm number is 10.

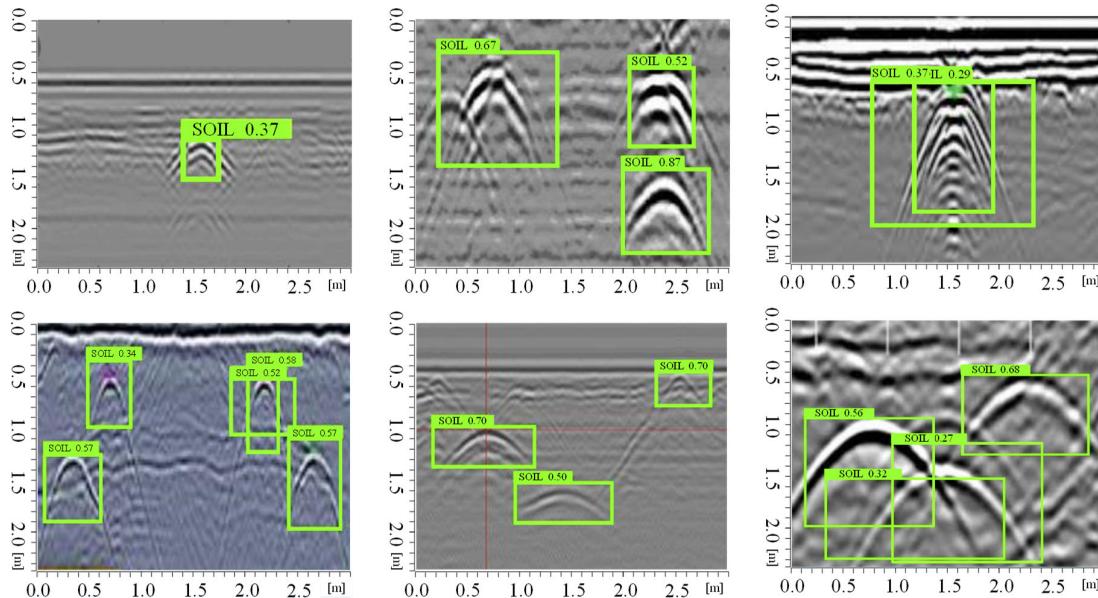


Figure 11. Single class targets detection performance

When there were multi-class targets in the detected GPR image, the predicted boxes can distinguish or identify the parabolas or voids. For some parabolas with multiple overlapping signals the vertex of curve was well positioned, as shown in Figure 12. Obviously, the less electromagnetic interference or noise in the GPR image, the better recognition and location performance. Those targets that are shallow from the soil surface had relatively obvious higher recognition scores. It can be seen that there were no misidentified boxes, all targets can be identified and located to the parabolic midpoint at overlapping positions. Figure 12 showed that the parabola with signal oscillation due to some highly conductive targets can be identified and located by the YOLO v3 detector with V-IoU. Overall, in this multi-class targets situation, total 82 hyperbolas in GPR image were tested. The correct detection number is 62, missed targets number is 4, and false alarm number is 5.

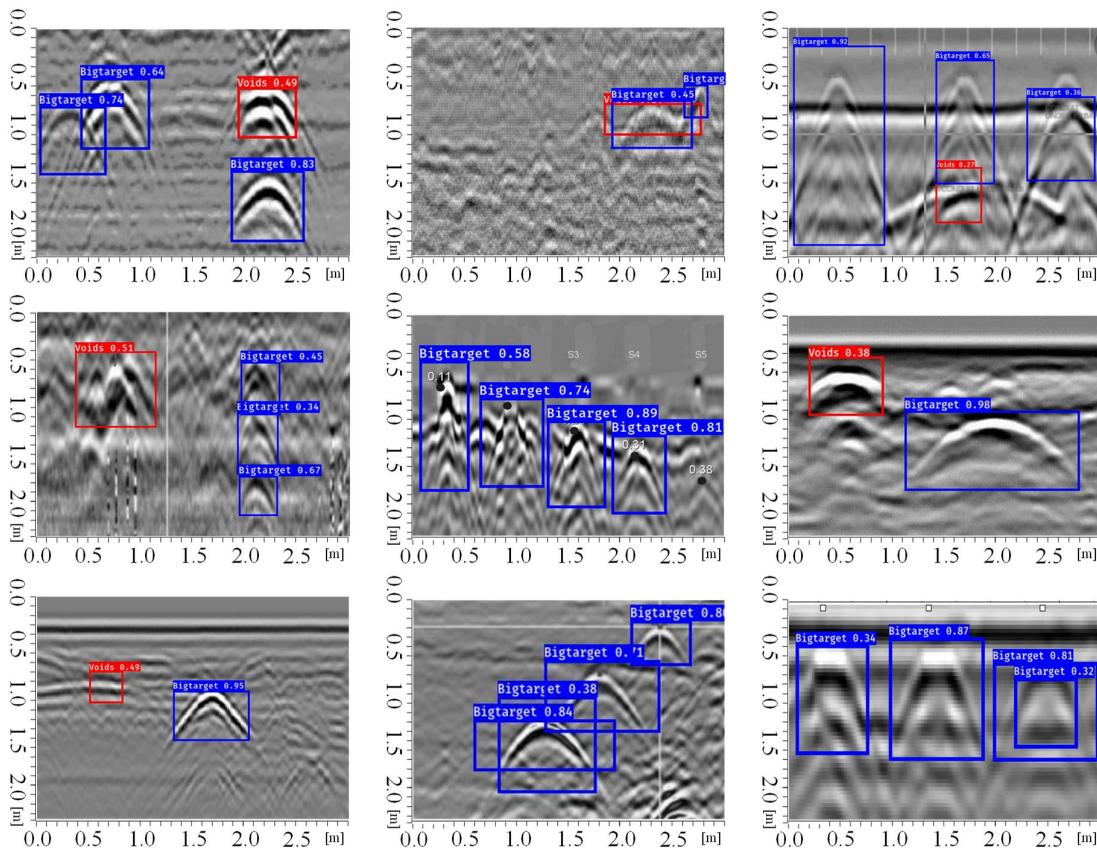


Figure 12. Multi-class class targets detection performance.

In engineering applications, we often need to detect the number of metal bars among concrete structures. It can be seen from Figure 13 that for the number of single-layer steel bars, the predicted bounding boxes can be positioned accurately; but for the multi-layer-reinforced concrete structure, there exists a case of missing identification. After many experiments and data statistics, if taking the number of hyperbola as a performance index, the YOLO V3 artificial intelligence recognition method proposed in this paper can predict the number of ground truth targets in GPR image by 90% accuracy, and its position error is less than 10% length unit. When detecting the number of concrete structures, total 192 hyperbolas in GPR image were tested. The correct detection number is 175, missed targets number is 11, and false alarm number is 8. Overall, YOLO v3 can achieve satisfactory performance when recognizing and positioning electromagnetic wave from GPR image features.

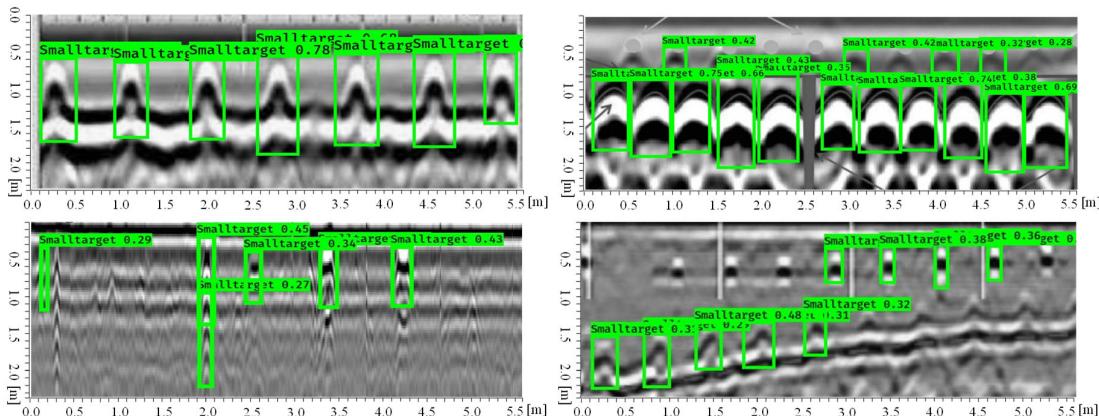


Figure 13. Pattern recognition of densely distributed reinforced concrete structures.

3.5. Learning Rate and Mean Average Precision Comparison

The learning rate directly affects the convergence state of the YOLO v3 training performance, and batch size affects the generalization performance. Earlier, we have discussed the Adam adaptive algorithm to update the global learning rate. In TensorFlow, we set the initial parameters of the learning rate to the same value. Here we evaluate the model optimization of SSD, faster-rcnn, and VIoU-YOLO v3 through the change of learning rate in training epoch. As shown in Figure 14, the learning rates of SSD, faster-rcnn, and VIoU-YOLO v3 were between 52 and 72 in epoch. The YOLO v3 has converged to a stable value when epoch was 73, which made the updated weight of loss value in TensorFlow to be reduced to the global threshold in a shorter time. The change of SSD is very close to VIoU-YOLO v3, but we can see from Figure 15 that the same situation occurs again similarly to Figure 9. Loss value of the SSD model will easily converge to its local optimal value with the increase of training times; obviously, after comparing the learning rate and loss value, the convergent speed of YOLO v3 with VIoU is more ideal.

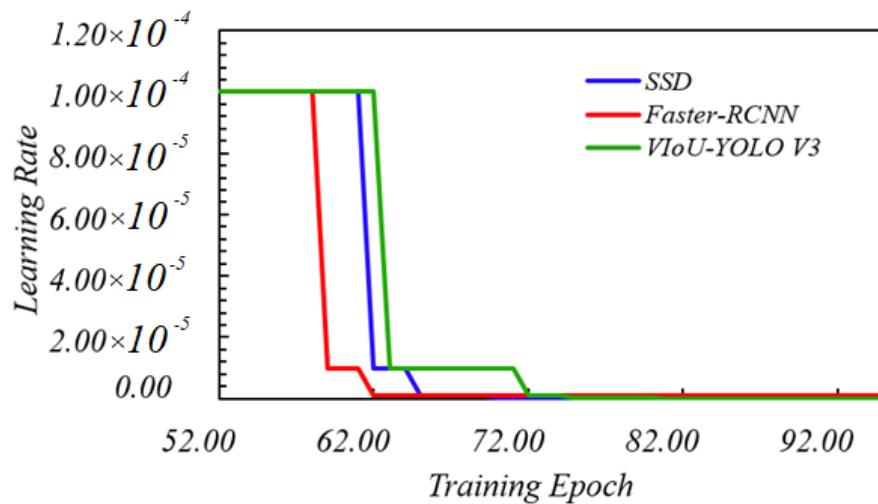


Figure 14. Learning rate versus training epoch.

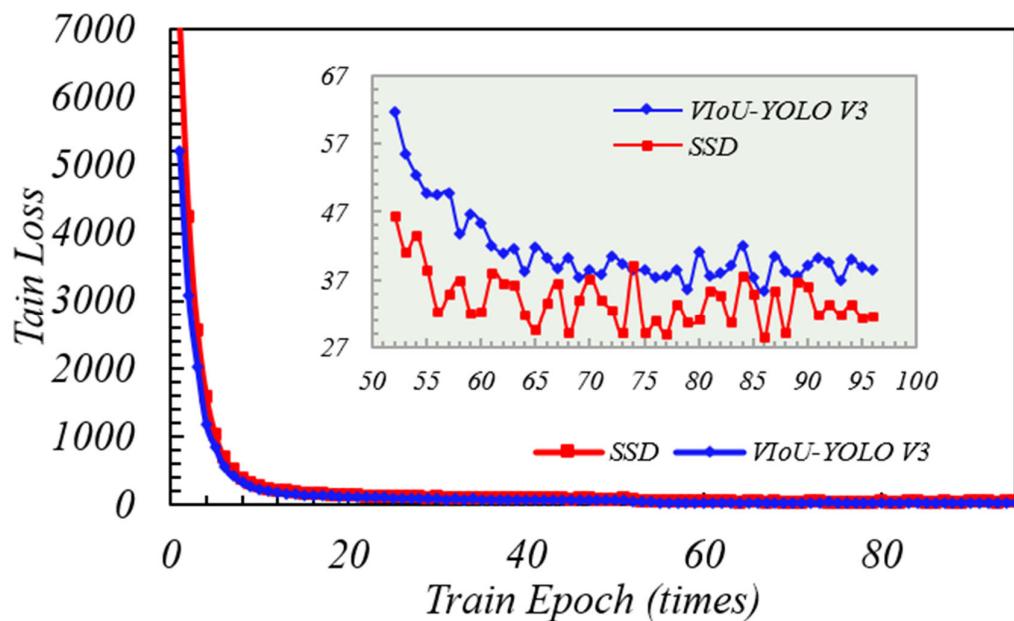


Figure 15. Loss versus training epoch.

Furthermore, we compared mAP of SSD, faster-rcnn, YOLO v2 and YOLO v3 with different V-IoU (or IoU) thresholds and scenes. We used 300 GPR image samples to generate Table 1. Here, the mAP_{50} means its IoU threshold preset as 0.5 and mAP_{75} preset as 0.75. Similarly, the mAP_{sc} , mAP_{mc} , and mAP_{metal_bars} represents the single classification, multi-class targets detection and only contains single layer metal bars scenes respectively. As shown in Table 1, after comparison, when the V-IoU threshold was 0.50, YOLO v3 with darknet-53 as the backbone can achieve a maximum mAP of 83.16; the SSD with ResNet-34 as the backbone can achieve an mAP of 75.66. The mAP scores of Faster-RCNN, YOLO v2, and v3 are more or less. When the IoU threshold was 0.75, the mAP scores of YOLO v3 and VIoU YOLO v3 are 77.15 and 75.90, respectively; SSD achieved a maximum mAP score of 79.80. In the detection which have multi-classes targets of GPR image, it is clear that YOLO v3 achieved an ideal mAP score. Comparing the mAP score of single class scenes, V-IoU YOLO v3 scored 83.17; in addition, when detecting the metal bars underground, although YOLO v3 achieved the highest mAP score of 79.90, V-IoU YOLO v3 still scored 76.10. In general, V-IoU YOLO v3 can achieve the best performance for three different real-time scenes.

Table 1. mAP comparison with five detection algorithms.

Algorithm	Backbone	mAP_{50}	mAP_{75}	mAP_{sc}	mAP_{mc}	mAP_{metal_bars}
SSD	ResNet-34	75.66	79.80	79.37	71.44	66.31
Faster-RCNN	ResNet-18	81.45	66.22	74.21	77.09	68.51
YOLO v2	Darknet-19	80.34	72.05	80.08	66.15	68.92
YOLO v3	Darknet-53	83.16	77.15	85.82	76.30	79.90
VIoU-YOLO v3	Darknet-53	82.71	75.90	84.56	83.17	76.10

3.6. Real-Time Performance and fps Testing

In expectation of testing the real-time detection speed of YOLO v3, we randomly selected five batches from 331 GPR images with size 416×416 ; the number of image batches were 100, 150, 200, 250, and 300, respectively, and took the mAP value in Table 1 as reference. Computer processor still is Intel(R) Gold 6130 with CPU with 2.10 GHz. As shown in Figure 16, when the batch size was 200, the detection speed of SSD can reach to 11 fps. After comparison, the detection speed of Faster-RCNN in each batch was not ideal, and its maximum detection speed is just 5fps. It can be seen from Figure 16 that the average detection speed of YOLO v2 is 5 fps. The fastest detection speed of YOLO v3 and VIoU-YOLO v3 can reach 15fps, and their average value is around 12fps. In other words, when the vehicle is equipped with GPR device, its detection speed can reach between 10 km/h and 20 km/h. Consequently, the VIoU-YOLO v3 detection method proposed in this paper can fulfill the real-time detection requirements.

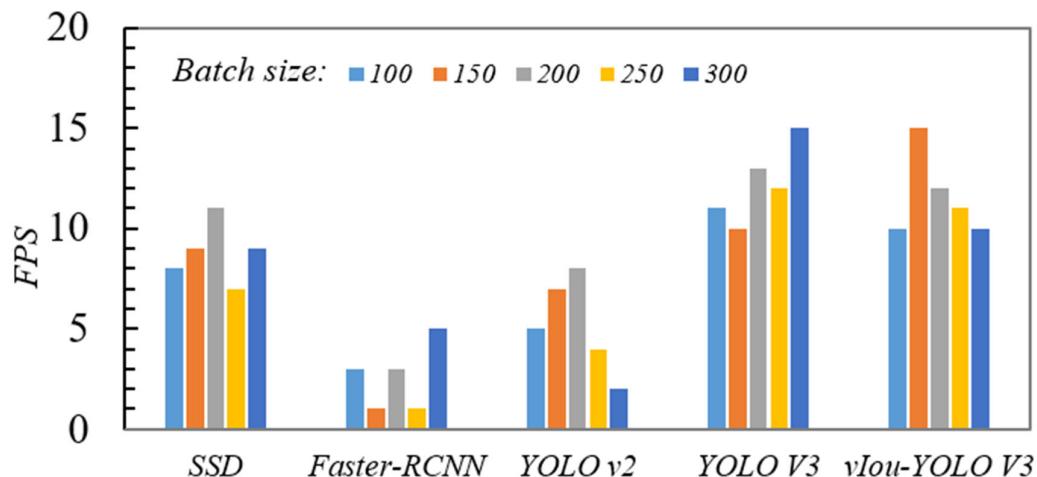


Figure 16. The fps versus batch size in five detection algorithms.

4. Conclusions

In this paper, a YOLO v3 was applied to build neural network detector to achieve real-time pattern-recognition of GPR images. It can be applied to actual underground detection engineering with meaningful accuracy and robustness based on Tensorflow, but this article is also limited to less samples and detection types of targets. Overall, this paper developed an innovative research application based on artificial intelligence algorithm in the field of electromagnetic wave detection. The main conclusions are as follows:

- (1) Redefined the encode approach of YOLO v3 and proposed a labeling technique with using parabolic vertices as feature points; this provides a high-precision encoding technique for locating targets in GPR image.
- (2) Proposed the principle of V-IoU; when the position of parabola vertex is within a certain range, free the limitation of IoU threshold. This method effectively reduces the false recognition rate caused by electromagnetic interference.
- (3) The V-IoU-YOLO v3 neural network can achieve 83.17 mAP score in the single class pattern-recognition scenes and 76.10 mAP score when detecting metal bars in concrete structures.
- (4) The VIoU-YOLO v3 detecting speed can reach 15fps under the CPU processor, and this speed can meet the real-time operation requirements of vehicle equipped with GPR device.

Author Contributions: Conceptualization, Z.Z.; validation, Y.L. (Yangfan Luo), and Y.L. (Yuanhong Li); formal analysis, Y.L.; investigation, Z.Q.; writing—original draft preparation, Y.Z.; writing—review and editing, Z.Z., Y.L. (Yuanhong Li), Z.Q.; funding acquisition, Z.Z. All authors have read and agreed to the published version of the manuscript.

Funding: The authors would like to acknowledge the support of this study by the State Key Research Program of China (Grant No. 2016YFD0700101), the State Key Research Program of China (Grant No. 2017YFD0700404), the Guangdong Provincial Department of Agriculture's Specialized Program for Rural Area Rejuvenation(Grant No. 2019KJ129), and the Guangdong Provincial Department of Agriculture's Modern Agricultural Innovation Team Program for Animal Husbandry Robotics (Fund No. 200-2018-XMZC-0001-107-0130).

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Sato, M.; Fujiwara, J.; Feng, X.; Zhou, Z.S.; Kobayashi, T. GPR development for landmine detection. *Geophys. and Geophys. Explorat.* **2005**, *8*, 270–279.
2. Li, Y.; Zhao, Z.; Xu, W.; Liu, Z.; Wang, X. An effective FDTD model for GPR to detect the material of hard objects buried in tillage soil layer. *Soil Tillage Res.* **2019**, *195*, 104353, doi:10.1016/j.still.2019.104353.
3. Zhang, J.; Huang, M.; Jin, X.; Li, X. A Real-Time Chinese Traffic Sign Detection Algorithm Based on Modified YOLOv2. *Algorithms* **2017**, *10*, 127, doi:10.3390/a10040127.
4. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. Ssd: Single shot multibox detector. In Proceedings of the 8th European Conference on Computer Vision, Amsterdam, The Netherlands, 12–16 October 2016; Springer: Cham, Switzerland, 2016.
5. Sonoda, J.; Kimoto, T. Object Identification form GPR Images by Deep Learning. In Proceedings of the 2018 Asia-Pacific Microwave Conference (APMC), Kyoto, Japan, 6–9 November 2018; Institute of Electrical and Electronics Engineers (IEEE): Piscataway, NJ, USA 2018; pp. 1298–1300.
6. Aydin, E.; Yüksel, S.E. May. Buried target detection with ground penetrating radar using deep learning method. In 2017 25th Signal Processing and Communications Applications Conference (SIU); Antalya, Turkey, 15–18 May 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 1–4.
7. Dinh, K.; Gucunski, N.; Duong, T.H. An algorithm for automatic localization and detection of rebars from GPR data of concrete bridge decks. *Autom. Constr.* **2018**, *89*, 292–298, doi:10.1016/j.autcon.2018.02.017.
8. Pham, M.-T.; Lefèvre, S. Buried Object Detection from B-Scan Ground Penetrating Radar Data Using Faster-RCNN. In Proceedings of the IGARSS 2018 - 2018 IEEE International Geoscience and Remote Sensing Symposium, Valencia, Spain, 22–27 July 2018; Institute of Electrical and Electronics Engineers (IEEE): Piscataway, NJ, USA, 2018; pp. 6804–6807.

9. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*; Neural Information Processing Systems Foundation Inc.: San Diego, CA, USA, 2015; pp. 91–99.
10. Girshick, R. Fast R-CNN. In Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015; pp. 1440–1448.
11. He, K.; Gkioxari, G.; Dollar, P.; Girshick, R. Mask R-CNN. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 2980–2988.
12. Kechagias-Stamatis, O.; Aouf, N.; Belloni, C. SAR automatic target recognition based on convolutional neural networks. In Proceedings of the International Conference on Radar Systems (Radar 2017), Belfast, UK, 23–26 October 2017.
13. Dou, Q.; Wei, L.; Magee, D.R.; Cohn, A.G. Real-Time Hyperbola Recognition and Fitting in GPR Data. *IEEE Trans. Geosci. Remote. Sens.* **2016**, *55*, 51–62, doi:10.1109/tgrs.2016.2592679.
14. Pham, M.-T.; Courtrai, L.; Friguet, C.; Lefèvre, S.; Baussard, A. YOLO-Fine: One-Stage Detector of Small Objects under Various Backgrounds in Remote Sensing Images. *Remote. Sens.* **2020**, *12*, 2501, doi:10.3390/rs12152501.
15. Redmon, J.; Farhadi, A. Yolov3: An incremental improvement. *arXiv preprint* **2018**, arXiv:1804.02767.
16. Peng, X.; Schmid, C. Multi-Region Two-Stream R-CNN for Action Detection. In *Proceedings of the Public-Key Cryptography—PKC 2018*; Springer: Berlin/Heidelberg, Germany, 2016; pp. 744–759.
17. Vinayakumar, R.; Alazab, M.; Soman, K.P.; Poornachandran, P.; Al-Nemrat, A.; Venkatraman, S. Deep Learning Approach for Intelligent Intrusion Detection System. *IEEE Access* **2019**, *7*, 41525–41550, doi:10.1109/access.2019.2895334.
18. Benjdira, B.; Khursheed, T.; Koubaa, A.; Ammar, A.; Ouni, K. Car Detection using Unmanned Aerial Vehicles: Comparison between Faster R-CNN and YOLOv3. In Proceedings of the 2019 1st International Conference on Unmanned Vehicle Systems-Oman (UVS), Muscat, Oman, 5–7 February 2019; Institute of Electrical and Electronics Engineers (IEEE): Piscataway, NJ, USA, 2019; pp. 1–6.
19. Betti, A.; Michelozzi, B.; Bracci, A.; Masini, A. Real-Time target detection in maritime scenarios based on YOLOv3 model. *arXiv Prepr.* **2020**, arXiv:2003.00800.
20. Prihatmaja, P.A.; Widiantoro, D.H. Improving Performance of YOLOv3 for Vehicle Detection. In Proceedings of the 2019 International Conference of Advanced Informatics: Concepts, Theory and Applications (ICAICTA), Yogyakarta, Indonesia, 20–21 September 2019; Institute of Electrical and Electronics Engineers (IEEE): Piscataway, NJ, USA, 2019; pp. 1–6.
21. Ju, M.; Luo, H.; Wang, Z.; Hui, B.; Chang, Z. The Application of Improved YOLO V3 in Multi-Scale Target Detection. *Appl. Sci.* **2019**, *9*, 3775, doi:10.3390/app9183775.
22. Lee, J.-S.; Yu, J.-D. Non-destructive Method for Evaluating Grouted Ratio of Soil Nail Using Electromagnetic Wave. *J. Nondestruct. Evaluation* **2019**, *38*, 41, doi:10.1007/s10921-019-0582-9.
23. Li, Y.; Wang, X.; Zhao, Z.; Han, S.; Liu, Z. Lagoon water quality monitoring based on digital image analysis and machine learning estimators. *Water Res.* **2020**, *172*, 115471, doi:10.1016/j.watres.2020.115471.
24. Zhou, L.; Zhou, L.; Wang, Z.; Wang, X. Soil Water Content Estimation Using High-Frequency Ground Penetrating Radar. *Water* **2019**, *11*, 1036, doi:10.3390/w11051036.
25. Qurishee, M.A. *Low-Cost Deep Learning UAV and Raspberry Pi Solution to Real Time Pavement Condition Assessment*; University of Tennessee at Chattanooga: Chattanooga, Tennessee, 2019.
26. Arora, P.; Deepali; Varshney, S. Analysis of K-Means and K-Medoids Algorithm for Big Data. *Procedia Comput. Sci.* **2016**, *78*, 507–512, doi:10.1016/j.procs.2016.02.095.
27. Rajiv, K.; Chandra, G.R.; Rao, B.B. GPR objects hyperbola region feature extraction. *Adv. Comp. Sci. Techn.* **2017**, *10*, 789–804.
28. Yu, S.-S.; Chu, S.-W.; Wang, C.-M.; Chan, Y.-K.; Chang, T.-C. Two improved k-means algorithms. *Appl. Soft Comput.* **2018**, *68*, 747–755, doi:10.1016/j.asoc.2017.08.032.
29. Liu, X.; Gao, W.; Zhu, X.; Li, M.; Wang, L.; Zhu, E.; Liu, T.; Kloft, M.; Shen, D.; Yin, J. Multiple Kernel k-means with Incomplete Kernels. *IEEE Trans. Pattern Anal. Mach. Intell.* **2019**, *42*, 1, doi:10.1109/tpami.2019.2892416.
30. Lee, E.; Schmidt, M.; Wright, J. Improved and simplified inapproximability for k-means. *Inf. Process. Lett.* **2017**, *120*, 40–43, doi:10.1016/j.iplet.2016.11.009.
31. Jeong, Y.; Lee, J.; Moon, J.; Shin, J.H.; Lu, W. K-means Data Clustering with Memristor Networks. *Nano Lett.* **2018**, *18*, 4447–4453, doi:10.1021/acs.nanolett.8b01526.

32. Hosang, J.; Benenson, R.; Schiele, B. Learning non-maximum suppression. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 4507–4515.
33. Rezatofighi, H.; Tsoi, N.; Gwak, J.; Sadeghian, A.; Reid, I.; Savarese, S. Generalized Intersection Over Union: A Metric and a Loss for Bounding Box Regression. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; Institute of Electrical and Electronics Engineers (IEEE): Piscataway, NJ, USA, 2019; pp. 658–666.
34. Rahman, A.; Wang, Y. Optimizing intersection-over-union in deep neural networks for image segmentation. In International Symposium on Visual Computing, Las Vegas, NV, USA, 12–14 December 2016; Springer: Cham, Switzerland, 2016; pp. 234–244.
35. Kim, K.-J.; Kim, P.-K.; Chung, Y.-S.; Choi, D.-H. Performance Enhancement of YOLOv3 by Adding Prediction Layers with Spatial Pyramid Pooling for Vehicle Detection. In Proceedings of the 2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), Auckland, New Zealand, 27–30 November 2018; Institute of Electrical and Electronics Engineers (IEEE): Piscataway, NJ, USA, 2018; pp. 1–6.
36. Ishimaru, A. *Electromagnetic Wave Propagation, Radiation, and Scattering: From Fundamentals to Applications*; John Wiley & Sons: Hoboken, NJ, USA, 2017.
37. Mao, Q.-C.; Sun, H.-M.; Liu, Y.-B.; Jia, R.-S. Mini-YOLOv3: Real-Time Object Detector for Embedded Applications. *IEEE Access* **2019**, *7*, 133529–133538, doi:10.1109/access.2019.2941547.
38. Adarsh, P.; Rathi, P.; Kumar, M. YOLO v3-Tiny: Object Detection and Recognition using one stage improved model. In Proceedings of the 2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS), Coimbatore, India, 6–7 March 2020; Institute of Electrical and Electronics Engineers (IEEE): Piscataway, NJ, USA, 2020; pp. 687–694.
39. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv preprint* **2014**, arXiv:1412.6980.
40. Kim, D.H. Evaluation of COCO Validation 2017 Dataset with YOLOv3. *Evaluation* **2019**, *6*, 10356–10360.
41. Zhang, Z.; He, T.; Zhang, H.; Zhang, Z.; Xie, J.; Li, M. Bag of freebies for training object detection neural networks. *arXiv Prepr.* **2019**, arXiv:1902.04103.
42. Ammar, A.; Koubaa, A.; Ahmed, M.; Saad, A. Aerial Images Processing for Car Detection using Convolutional Neural Networks: Comparison between Faster R-CNN and YoloV3. *arXiv Prepr.* **2019**, arXiv:1910.07234.
43. Aranganayagi, S.; Thangavel, K. Clustering Categorical Data Using Silhouette Coefficient as a Relocating Measure. In Proceedings of the International Conference on Computational Intelligence and Multimedia Applications (ICCIMA 2007), Sivakasi, India, 13–15 December 2007; Institute of Electrical and Electronics Engineers (IEEE): Piscataway, NJ, USA, 2007; Volume 2, pp. 13–17.
44. Yue, Y.; Finley, T.; Radlinski, F.; Joachims, T. A support vector method for optimizing average precision. In Proceedings of the Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval—SIGIR ’07, Amsterdam, The Netherlands, 23–27 July 2007; Association for Computing Machinery (ACM): New York, NY, USA, 2007; pp. 271–278.
45. Gour, D.; Kanskar, A. Optimised YOLO: Algorithm for CPU to Detect Road Traffic Accident and Alert System. International. *J. Eng. Res. Technol.* **2019**, *8*, 160–163.

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).