Bayer Marker Detection based on Yolov3

Xingxing Li
Department of electronics and information engineering, Guangzhou College of Technology and Business, FoShan, 528138, China
lixingxing_Edooy@163.com

Abstract. With the development of artificial intelligence technology, target detection is widely used in many commercial fields, such as automatic driving, security monitoring and robotics, especially the deep learning algorithm has brought about a significant improvement in target detection performance. Bay number identification has a wide range of applications in industrial scene understanding. It can be used for character positioning and is applied to driverless. The accurate detection of shell number is the basis for subsequent character recognition. Character recognition is after character detection. In this paper, yolov3 target detection algorithm is used to realize the shell number text detection. Compared with yolov2 algorithm, yolov3 uses multi-scale prediction, and adopts feature maps of different scales to extract image features to adapt to different sizes of objects, and to use better. The basic classification network (ResNet-like) and classifiers use multiple logistic classifiers instead of the sorttmax classifier to solve the problem of multi-label classification.

1. Introduction
In recent years, target detection is one of the important tasks of computer vision. Recently, deep learning has made great breakthroughs in the field of image classification. Researchers naturally think of converting target detection tasks into image classification problems and have achieved good performance. In the detection system, an image is evaluated by using a classifier, and whether a window or a bounding box in a certain area of each position contains an object containing the object. Some detection systems, such as Deformable Parts Models (DPM), use a sliding window approach. The recent R-CNN and Fast R-CNN use the region proposals method [1], and some potential bindings that may contain objects. Box, then through a classifier to determine whether the bounding box contains objects, and the possibility and confidence of the category of the object, and Faster R-CNN further cancels the process of extracting the region, and introduces the full volume of the extracted region. The product network (RPN) (Region Proposal Network), and share the convolution feature of the full map with the target detection network, greatly reducing the time consumption of calculating the region, but these methods need to go through several independent parts, so the detection speed will be slow, also It is difficult to optimize because each individual part needs to be trained separately. YOLO model first converts the input image into a network of 448×448 size, and the model divides the image into S×S grids, and lets each grid predict B bounding boxes. In the training phase, for a target, if the center position of the target falls in a certain grid, then the grid is responsible for finding the target, that is, positioning the target. The bounding box predicted by each grid includes five kinds of information, namely the target abscissa x, the ordinate y, the width w, the height h, and the confidence level. If the predicted bounding
box does not contain the target, the confidence is set. If it is 0, if the bounding box contains the target completely, the confidence is 1. If only part of the target is included, the confidence is determined by the IOU containing the target degree, and each grid also predicts the probability of occurrence of C categories. [2].

Yolov2 network adds batch normalization after each convolutional layer, and it is not easy to over-fitting after dropping the dropout. At the same time, the network raised the resolution to 448×448. The previous yolo used the data of the fully connected layer to complete the prediction of the border, resulting in the loss of more spatial information and inaccurate positioning. Therefore, the author draws on the anchor idea in Faster R-CNN, which is a key step in the RNP network. It is a sliding window operation on the convolution feature map. Each center can predict 9 different sizes of suggestion boxes. After joining the anchor boxes, the accuracy rate dropped slightly and the recall rate increased by 7%. Another improvement, unlike the previous selection of boxes dimensions, the author used the K-means clustering method to train bounding boxes, which can automatically find better boxes wide and high dimensions [3].

The biggest change in yolov3 over the previous network includes two points: using the residual model and adopting the FPN architecture. The feature extractor of yolov3 is a residual model. Because it contains 53 convolutional layers, it is called Darknet-53. From the network structure, the residual unit is used compared to the Darknet-19 network, so it can be constructed deeper. Another point is to use the FPN architecture (Feature Pyramid Networks for Object Detection) to achieve multi-scale detection. Yolov3 uses a three-scale feature map (when the input is 416×416): (13*13), (26*26), (52*52), yolov3 uses 3 a priori boxes for each position, so use K-means gets 9 a priori boxes and divides them into 3 scale feature maps. The larger scale maps use smaller a priori boxes, similar to SSDs. At the same time, using logistic regression to regress the box confidence, the positive and the actual block IOU is greater than 0.5 as a positive example. Unlike SSD [6], if there are multiple a priori to meet the target, only one IOU maximum a priori is taken. Logistic regression is used independently for each category, and the two-class cross-entropy loss is used as a class loss, which can handle multi-label tasks well [4, 5].

2. Introduction of Methods

2.1. YOLOV3 backbone

Yolov3, compared to SSD [6], FasterRCNN, RetinaNet, the speed is faster, and there is no pooling layer and full connection layer in the whole v3 structure. The network structure of yolov3 is shown in Figure 1.
Figure 1. Yolov3 network structure.

In Figure 1, yolov3 uses "leaky ReLU" as the activation function, using batch normalization as a method of regularization, accelerated convergence, and avoidance of overfitting. After the BN layer and the leaky relu layer are connected to each layer of the convolutional layer. The basic component of yolov3 is convolution + BN + Leaky relu. The large component of yolov3 is the residual unit. Yolov3 begins to draw on ResNet's residual structure. Using the residual structure can make the network structure deeper (from darknet-19 of v2 to darknet-53 of v3). Yolov3 splicing the upper layer of the darknet and the next layer of the layer. The operation of the splicing is different from the operation of the residual layer add. The splicing will expand the dimension of the tensor, and the add is just a direct addition and will not cause Zhang. The change in the dimension.

Figure 2. Feature maps at three different scales
Yolov3 draws on the FPN (feature pyramid networks) feature pyramid network, uses multiple scales to detect targets of different sizes, and the network outputs three different scale feature maps, as shown in Figure 2, scale1, scale2, scale3, each The Scale layer first obtains the depth features of the image through the DBL basic CNN convolutional neural network feature extraction unit (convolution+BN+leaky relu), and then obtains a feature map of different scales through a convolution. The finer the grid cell is, the finer the object can be detected. The depths of scale1, scale2, and scale3 are both 255, and the rule of side length is 13:26:52, and each box should output a probability for each category. Yolov3 is set to predict 3 boxes per grid unit, so each box needs to have five basic parameters (x, y, w, h, confidence), because the category to be trained in this article is only one type of shell number. Therefore, there is only one category of probability, output 3*(5 + 1) = 18 parameters.

Yolov3 uses the up sampling method to implement this multi-scale feature map. It can be combined with the right side of Figure 1 and Figure 2. The two tensors of the concat connection in Figure 1 have the same scale (the two splicing are 26 * 26 scales respectively). Splicing and 52 * 52 scale splicing, through (2, 2) up sampling to ensure that the concat splicing tensor scale is the same), there is no SSD algorithm directly using the backbone intermediate layer processing results as the output of the feature map, but with the network layer behind The up sampled result is processed as a feature map after a stitching. On the one hand, avoiding overlap with other algorithmic practices, on the other hand, saving model parameters.

2.2. Bounding Box Prediction

The bounding Box Prediction in yoloV3 draws on the anchor mechanism in the faster R-CNN RPN, but there is no manual anchor prior (template box), so the method of dimension clustering is used to determine the anchor box prior, and finally the cluster is determined. Prior can also perform well at k=5, so k=5 is chosen, but due to the instability of the linear regression of the anchor mechanism (because the offset of the regression can make the box offset anywhere in the picture), the algorithm directly predicts The relative position is used to predict the relative coordinates of the b-box center point with respect to the upper left corner of the grid unit, using the following formula 1-4.

\[
b_x = \sigma(t_x) + c_x
\]

\[
b_y = \sigma(t_y) + c_y
\]

\[
b_w = p_w e^{t_w}
\]

\[
b_h = p_h e^{t_h}
\]

Yolov3 uses logistic regression, which is used to make an objective score on the part enclosed by the anchor, that is, how likely this position is to be the target. This step is done before the predict, you can remove the unnecessary anchor, you can reduce the amount of calculation. Where \(c_x, c_y\) is the coordinate offset of the network, \(p_w, p_h\) is the side length of the preset anchor box, and the resulting frame coordinate value is \(b_{(x, y, w, h)}\), and the network learning target is \(t_{(x, y, w, h)}\). Unlike the faster R-CNN, yolov3 only operates on one prior, which is the best prior. Logistic regression is the one used to find the highest objectness score from the nine anchor priors. Logistic regression is a linear modeling of the relationship between the prior and the objectness score.

2.3. Loss function

Yolov3 uses a sum-squared error, as shown in Equation 5:
\[
\text{Loss} = \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B^2} \left[ \left( \sigma(x_i^j) - \sigma(\hat{x}_i^j) \right)^2 + \left( \sigma(y_i^j) - \sigma(\hat{y}_i^j) \right)^2 \right] + \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B^2} \left[ (t_{x_i^j} - \hat{t}_{x_i^j})^2 + (t_{y_i^j} - \hat{t}_{y_i^j})^2 \right] + \\
\sum_{i=0}^{S^2} \sum_{j=0}^{B^2} \sum_{n=1}^{n_{\text{noobj}}} \left[ (\xi_{x_i^j}^n - \hat{\xi}_{x_i^j}^n)^2 + (\xi_{y_i^j}^n - \hat{\xi}_{y_i^j}^n)^2 \right] + \sum_{i=0}^{S^2} \sum_{j=0}^{B^2} \sum_{c=1}^{n_{\text{classes}}} (\alpha(c) - \hat{\alpha}(c))^2
\]

The first term in Equation 5 represents the predicted coordinate error, the second term represents the IOU error calculated online, and the third term represents the classification error. Only the mesh containing the target has a class loss, and the others do not need to return the loss. This formula calculates the loss value relative to the predicted bounding box position \((x, y)\), where \(\lambda\) is assumed to be a given constant. The formula calculates the sum of each of the bounding box prediction values \(j = (0, 1, \ldots, B)\) for each network element \((i = 0, 1, \ldots, S^2)\). \(t_{ij}^{\text{obj}}\) is defined as follows:

1. If there is a target in the mesh cell unit \(i\), the \(j\)th bounding box prediction value is valid for the prediction.
2. If there is no target in the mesh cell unit.

3. Experiments
The experimental environment is the Ubuntu system, which uses the 1080Ti GPU for training tests. The experimental data set is the 1050 shell number image captured by the camera at the port. The data set is divided into 8:2 ratios, of which 839 images are used as the training set, 211 images are used as the test set, and the images are labeled with tools. The algorithm is trained and iterated 4600 times. The training process is shown in Figure 3:

\textbf{Figure 3. Bay number model training}
As shown in Figure 3, the loss function decreases exponentially at the beginning of the training. When iterating to 1200 times, the loss rate decreases. Slowly descending from 0.2. At the same time, the detection and evaluation index MAP starts to increase exponentially with the number of iterations. When iterating to 1200 times, the speed is gradually slowed down, and the final recognition average accuracy is stable at about 99%.

This paper selects several representative images from the test set and uses the trained model weights to test. The test results are shown in Figure 4:

![Figure 4. Shell number detection effect.](image)

In Figure 4, the left half and the right half use different shooting angles. The left camera has a tilt angle, and the right camera looks down. The scale of the bay number is different from the light. Yolov3 target detection algorithm works well. Adapt to different sizes, scales, light changes, etc.

In this paper, the overall test is carried out on the test set using the trained weights. The effect of the model is shown in Table 1 below:

| Loss  | Iter  | MAP     | IOU     | Mem    | TP/FP/FN | Precision | Recall | F1-Score | FPS | BFLOPS |
|-------|-------|---------|---------|--------|----------|-----------|--------|----------|-----|--------|
| 0.026 | 5000  | 99.65%  | 88.7%   | 1509Mb | 361/24/0 | 94%       | 100%   | 97%      | 35  | 65.31  |

The model used in this paper is yolov3-obj. When iterating to 4600 times, the value of the loss function is reduced to 0.026, the MAP is 99.65%, the IOU is 88.7%, the model size is 1509 Mb, and the FPS is 35 frames. Yolov3 algorithm can accurately detect the initial shell number, and the algorithm can meet the requirements of real-time performance, which is the basic condition for the identification of the rear shell number.

4. Conclusion
Yolov3 uses residual units in the network structure, so it can be built deeper and adopts FPN architecture to achieve multi-scale detection. The algorithm uses three scale feature maps, and the larger scale maps use smaller a priori frames to accommodate different sizes of objects. At the same time, yolov3 uses logistic regression to regress the box confidence. As a positive example, the a priori and the actual block
IOU are greater than a certain threshold, and finally only one IOU maximum a priori. Logistic regression is used independently for each category. The two-class cross-entropy loss is used as the category loss, which can handle multi-label tasks well. The use of yolov3 to complete the shell number detection task can be used as a basis for real-time identification of the shell number.

Acknowledgments
This work was financially supported by fund project, that is, Guangzhou Institute of industry and commerce college level research project in 2019"Research and design of S band radio frequency front-end" KA201937.

References
[1] Ren S, He K, Girshick R, et al. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks [J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2015, 39 (6): 1137 - 1149.
[2] Redmon J, Divvala S, Girshick R, et al. You Only Look Once: Unified, Real-Time Object Detection [J]. 2015.
[3] Redmon J, Farhadi A. [IEEE 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) - Honolulu, HI (2017.7.21-2017.7.26)] 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) - YOLO9000: Better, Faster, Stronger [J]. 2017: 6517 - 6525.
[4] Redmon J, Farhadi A. YOLOv3: An Incremental Improvement [J]. 2018.
[5] Szegedy C, Ioffe S, Vanhoucke V, et al. Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning [J]. 2016.
[6] Liu W, Anguelov D, Erhan D, et al. SSD: Single Shot MultiBox Detector [C]// European Conference on Computer Vision. 2016.