

YOLO Sparse Training and Model Pruning for Street View House Numbers Recognition

by No No

Submission date: 01-Mar-2023 10:31PM (UTC-0600)

Submission ID: 2026756528

File name: datacenter_paper_turnitin_2023-03-02_13990708.docx (1.06M)

Word count: 3682

Character count: 19597

YOLO Sparse Training and Model Pruning for Street View House Numbers Recognition

Ruohao Zhang^{1,*}Yijie Lu²Zhengfei Song³

¹ Department of Software Engineering, Nanjing University

² Department of Artificial Intelligence, Nanjing University

³ School of Electronic and Information Engineering, Tongji University

*Corresponding author. Email: 201250203@smail.nju.edu.cn.

ABSTRACT

This paper proposes a YOLO (You Only Look Once) sparse training and model pruning technique for recognizing house numbers in street view images. YOLO is a popular object detection algorithm that has achieved state-of-the-art performance in various computer vision tasks. However, its large model size and computational complexity limit its deployment on resource-constrained devices such as smartphones and embedded systems.

To address this issue, we use a sparse training technique that trains YOLO with L1 norm regularization to encourage the network to learn sparse representations. This results in a significant reduction in the number of parameters and computation without sacrificing accuracy. Furthermore, we apply a model pruning technique to the sparse-trained model to further reduce the model size and computation. We evaluate our proposed method on the SVHN (Street View House Numbers) dataset and show that it achieves comparable performance to the original YOLO model while reducing the model size by 5% and the computation time by 7%. Overall, our proposed YOLO sparse training and model pruning technique provides an effective solution for deploying YOLO-based object detection models on resource-constrained devices.

Keywords: YOLO, sparse training, deep learning, model pruning.

1. INTRODUCTION

With the rapid development of intelligent technology, we have stepped into the era of intelligence, where a new intelligent way of life has become the trend of social development and real-time accurate recognition of characters is one of the key technologies. In recent years, Optical Character Recognition is a topic of wide interest in the field of computer vision and it has become the technical core of many new applications [1]. However, in real scene images such as the street view house numbers, characters are affected by many external factors, such as illumination, noise, and wear and tear, influencing the accuracy of target positioning and resulting in characters that are difficult to recognize or easily misidentified. Therefore, character recognition in real scenes still needs to be improved and the study of Optical Character Recognition in real scenes is undoubtedly meaningful and important.

Existing character recognition techniques can be broadly classified into two categories: recognition based on manual feature extraction and recognition based on machine learning. The former mainly uses feature extraction methods such as SIFT (Scale Invariant Feature Transform) and HOG (Histogram of Oriented Gradients), combined with classifiers to recognize characters [2][3]. This method is traditional, requires extensive processing of samples and is vulnerable to subjective factors. In contrast, the latter, recognition based on machine learning, uses machine learning methods such as CNN (Convolutional Neural Networks) or other networks to directly recognize and classify characters. The recognition effectiveness of this method is closely related to the size of the sample and the number of training sessions.

As computer vision technology has been developing rapidly and the collection of large amounts of data is becoming more and more easier, automatic feature extraction methods have become a hot topic of research, and researchers pay more attention to find ways to optimize this method to improve recognition efficiency and reduce training time. Due to the continuous research in deep learning, more models with better performance are proposed and put into use. This

paper, which is based on a discussion of theoretical knowledge related to deep learning networks, some mainstream models and their performance, focuses on YOLO v5 (You Only Look Once) and explores automatic recognition methods based on YOLO v5. The main research includes:

- (1) Briefly discuss and summarize the background and significance of this paper, and outline its research status in terms of the current status of recognition of street view house numbers, neural networks and deep learning.
- (2) Focus on YOLO v5 and the application of YOLO v5 to the street view house numbers recognition as well as using a dataset of street view house numbers images taken from real scenes as objects.
- (3) Research model optimization methods, experiment with pruning, and apply the pruning algorithms to YOLO v5 to improve the effectiveness of YOLO v5 models for street view house numbers recognition.

2. RELATED WORK

In the range of recognizing the street view house numbers recognition, there are several different ways that are worth being mentioned.

In the past, people used the classifier to classify the number into different categories and then translated the number string. E. Kussul et al. [4] first introduced the improved version of RSC classifier and perceptron, LIRA(Limited Receptive Area), with a view to ameliorate the classification speed. They deleted the GROUP layer of the RSC classifier. Instead, they replaced each pair neurons in the GROUP layer with just only one neuron which can represent either ON or OFF.

Next, people used CNN to make this process much more accurate and efficient. Chen, L. et al. [5] proposed an inventive way to increase the accuracy of CNN classifier, distorting the pictures. Since they thought that random distortion was contradictory with normalization and CNN was a very powerful tool, they chose random distortion to increase the accuracy of their model.

J. Redmon [6] proposed YOLO (You Only Look Once) as a tool to recognize the object in 2016, which brought about sensation at that time. YOLO is extremely fast as it predicts the result based on the entire image information. J. Redmon stated that YOLO was good at balancing the accuracy and efficiency. In most cases, the performance of YOLO is far better than all the methods such as Faster R-CNN that belong to the CNN family. Huang, R. [7] also believed that YOLO-LITE, another version of yolo used to work on the devices without GPU, is the best lightweight real-time object detection tool.

Furthermore, Cao, L.-C. et al. [8] not only used YOLO to locate the text but also proposed an algorithm to identify whether the text boxes that had been located were in the same line, which contributed to the higher accuracy and efficiency of handwritten text recognition.

However, an image segmentation may not be a must for digits recognition. Guo, Q. et al.[9] combined CNN with hidden Markov model(HMM) in a hybrid fashion to form the hybrid CNN-HMM architecture to solve street view number recognition problem. They transformed the problem into sequence recognition under probabilistic processing instead of identifying isolated characters after performing segmentation operations, saving a lot of labor in designing complex algorithms and feature labelling. Also, A.G. Hochuli [10] proposed a way to recognize numbers without segmentation algorithms. They created a framework based on four task-specific classifiers, one for estimating the number of touching components and three respectively recognizing [0-9],[00-99] and [000-999] numbers. With experiments on two large datasets, their work were proved to achieve SOTA.

When researchers used the data in the dataset to train the model, they might find some data might be damaged. When Shi S.-X. et al. [11] used PCA to train the model with some damaged data, they decided to repair the damaged picture. However, the results were not satisfying. As a result, they proposed the hypothesis that deleting these damaged data before training may be better.

3. DATABASE

3.1. Street View House Numbers Dataset

We use the Street View House Numbers Dataset (SVHN) in Google Street View Image, a real-world image dataset for developing machine learning and object recognition algorithms with minimal data processing and formatting requirements. It can be viewed as stylistically similar to MNIST dataset. Still, it contains an order of magnitude more

labeled data (over 600,000 digit images). It comes from a more complex, unsolved real-world problem (Recognizing Digits and Digits in Images of Natural Scenes).

The images in the SVHN dataset are all colour images with a size of 32x32 pixels and contain numbers from 1 to 5. Each number has a corresponding label, representing its value. The SVHN dataset consists of three parts: training set, test set, and validation set. The training set contains about 73,257 images, the test set contains approximately 26,032 images, and the supplementary set contains about 531,131 images.

Since the SVHN dataset is a real-world dataset, it has some unique properties. First, the numbers in the image may have different scales, rotations, distortions, and occlusions, which is a challenge for the training and testing of the number recognition model. Second, the distribution of digits in the SVHN dataset needs to be more balanced. Some digits appear more frequently than others, which may cause digit recognition models to perform poorly on some numbers. Therefore, when using the SVHN dataset for digit recognition tasks, some special pre-processing and data augmentation operations are required to improve the performance and robustness of the model.

The SVHN dataset has been widely used for training and testing digit recognition models and strongly influences academia and industry. It is a crucial benchmark dataset in digit recognition, which can be used to evaluate and compare the performance of different digit recognition models and algorithms.

3.2. Deal with the dataset

First, we need to read the JSON file, which contains the digital tag, number box location, and other information corresponding to each picture in the training set. Then, we load each image and get the width and height of the loaded image for subsequent normalization operations. Next, we get each number's label and location information in each picture. After that, we normalize the position information of the number box, that is, the coordinates and width of the upper left corner of each numeric box divided by the width and height of the image for subsequent model training. In the end, we write the processed digital label and normalized number box location information to a txt file named after the image file. This process can help us convert the dataset to the proper format for the YOLOv5 model.

4. YOLOV5 MODEL

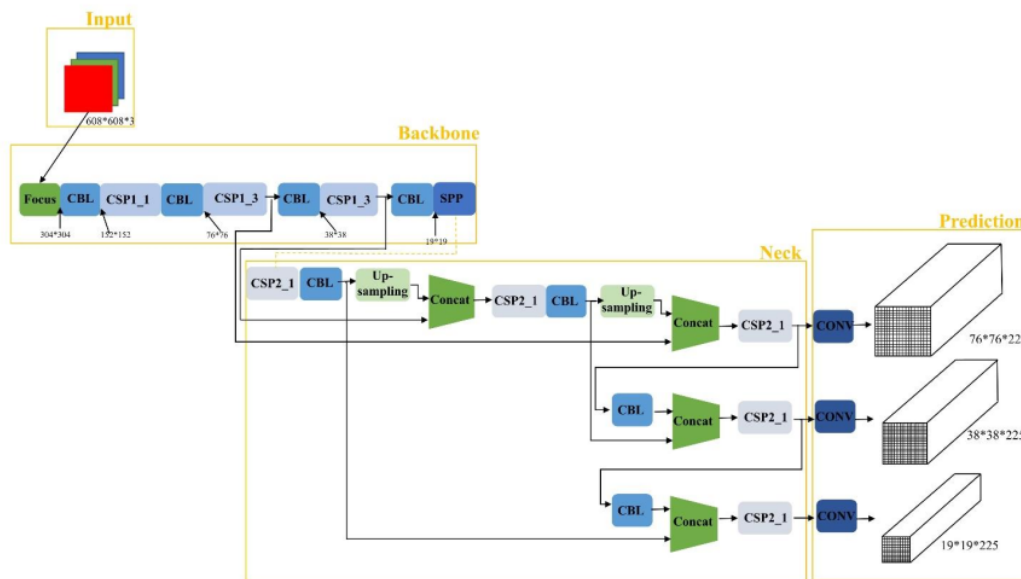


Figure 1 Structure of YOLO v5

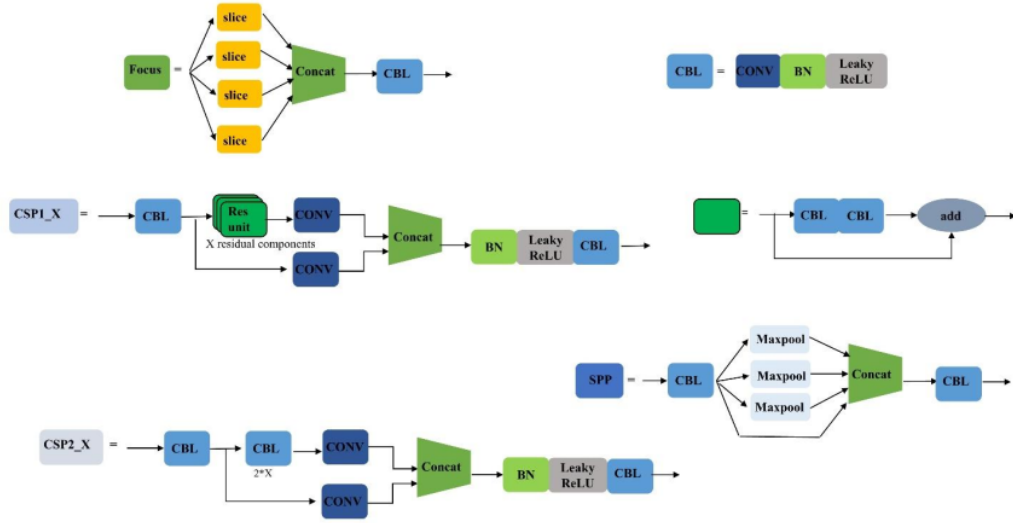


Figure 2 Structure of small parts in YOLO v5

The input side of YOLOv5 adopts Mosaic data enhancement, which uses 4 images, randomly scaled, cropped and arranged to stitch. It solves the problem of large number of small targets and unbalanced ratio of small to medium-sized targets in the dataset, helping to enrich the dataset and reduce the GPU at the same time. The input side also performs adaptive anchor frame calculation. During the network training, the network outputs the predicted frame on the basis of the initial anchor frame, then compares it with the real frame to calculate the difference between the two, and iterates the network parameters by backward updating. In addition, adaptive image scaling is also performed to uniformly scale images of different lengths and widths to the same standard size before sending them to the detection network.

The Backbone is firstly sliced in the Focus structure, where the original 608*608*3 image is sliced into 304*304*12 characteristic pattern, and then convolved with 32 convolution kernels once, and finally into 304*304*32 characteristic pattern. Subsequently, the CSP structure CSP1_X is built based on the CSPNet (Cross Stage Partial Network), which enables the convolutional neural network to maintain accuracy and reduce computation and memory costs while keeping lightweight.

The Neck part is the structure of FPN+PAN, where the FPN layer conveys strong semantic features from top down, and a bottom-up feature pyramid is added after the FPN layer to convey strong localization features upward. Also this part adds the CSP2 structure to enhance the network feature fusion.

The Prediction part uses CIOU_Loss as the loss function of the Bounding box.

$$CIOU_Loss = 1 - CIOU \quad (1)$$

$$CIOU = IOU - \frac{Distance_2^2}{Distance_C^2} - \frac{v^2}{(1 - IOU) + v} \quad (2)$$

$$IOU = \frac{\text{area of intersection of prediction frame and target frame}}{\text{area of concatenation of prediction frame and target frame}} \quad (3)$$

v is a parameter that measures the consistency of the aspect ratio,

$$v = \frac{4}{\pi^2} \left(\arctan \frac{w^{gt}}{h^{gt}} - \arctan \frac{w^p}{h^p} \right) \quad (4)$$

Distance_C is the diagonal distance of the minimum outer rectangle of the prediction frame and the target frame, and Distance_2 is the Euclidean distance of the two centroids.

CIOU_Loss takes the overlap area, centroid distance, and aspect ratio into account, making the prediction frame regression faster and more accurate. The nms non-maximum suppression operation is subsequently performed for many target frame filters.

5. PRUNE

5.1. Sparse Training

When building neural networks, the huge number of parameters can significantly slow down the training process. To speed up training and deployment, we implemented sparse learning techniques to compress our model and accelerate training and inference.

The core idea of sparse training is setting some neuron weight values to zero, making the model sparser. This decreases the number of parameters and calculations needed, reducing storage requirements. Specifically, we applied an L1 norm regularization approach to constrain weight values. We also pruned unimportant channels to further minimize parameters.

By implementing these techniques, we were able to optimize our model size, training time, and inference speed. Sparse training and channel pruning are useful tools for efficiently constructing and deploying deep neural networks. Streamlining networks in this way makes them more practical for real-world use. By trimming away unnecessary parameters and channels, we achieved a model that trains and infers quickly without sacrificing performance. Sparse training methods thus facilitate the development of deep learning models that strike a good balance between accuracy and efficiency.

5.2. BN Pruning

Batch Normalization (41) is a normalization technology commonly used in deep learning, which can accelerate convergence and improve the model's generalization performance. However (2), as the model size increases, the calculation amount of the BN layer also increases, and the storage and reasoning cost of the model also increases. In order to reduce the size of the model and accelerate model reasoning, BN-layer pruning has become a practical technology.

The principle of BN-layer pruning is to find channels or neurons that have less impact on the model's accuracy by analyzing the statistical information of the BN layer and deleting them from the network (4). The importance of each channel or neuron in the BN layer is obtained by analyzing the mean (4) and standard deviation of the BN layer; that is, the mean and standard deviation of the BN layer are obtained through the forward propagation (41) of the training set, and then the importance of each channel or neuron is determined according to the size of the mean and standard deviation.

The pruning of the BN layer can be divided into the following steps:

1. Calculate the importance of each channel or neuron in the BN layer: Calculate the significance of each channel or neuron according to the mean and standard deviation of the BN layer of the training set, and then sort it according to the importance.
2. Delete low-importance channels or neurons: Remove less important channels or neurons from the BN layer and delete the related weights accordingly.
3. Fixed pruning model: After deleting channels or neurons, the remaining models need to be fine-tuned to restore accuracy, and the model needs to be compressed and accelerated.

There are many ways to prune the BN layer, the most common of which is to prune based on the mean and standard deviation of the BN layer and to delete low-importance channels or neurons in the BN layer through thresholds. In addition, there is a BN layer pruning method based on sparse matrix and binarization, which can further reduce the model's storage and calculation and improve the model's running speed.

In our program, we use the method based on gradient information to assess the importance of the convolutional nucleus and then selectively prune the convolutional nucleus according to the assessment results.

In our experiment, the gradient information of all convolutional nuclei is obtained by training the complete YOLOv5 model on the training set. Then, the importance is assessed by calculating the average gradient size of each convolutional kernel. The smaller the average gradient, the smaller the impact of the convolutional kernel on the model; it can be pruned. Then, YOLOv5prune uses a method based on L1 regularization to prune the convolutional nucleus selectively, that is, the convolutional nucleus with a gradient value less than a certain threshold, while retaining an essential convolutional nucleus so that the model accuracy loss after pruning is less.

6. EXPERIMENTS AND DISCUSSION

The model we used to detect digits was trained on a GeForce RTX 2070 SUPER. We used pytorch 1.21.1+cu116 and Cuda 11.6.

The main indicators we used to evaluate our model were Intersection over union(IOU), Precision(P), Recall(R) and mean of Average Precision(mAP). Precision illustrates how accurate the model is. Recall illustrates how the model can make correct predictions among all ground truth positive examples. Mean of Average Precision stands for the overall accuracy of the model.

$$P = \frac{TP}{TP + FP} \times 100\% \quad (5)$$

$$R = \frac{TP}{TP + FN} \times 100\% \quad (6)$$

$$AP = \int_0^1 P(R) dR \quad (7)$$

$$mAP = \frac{\sum_{i=1}^N AP_i}{N} \quad (8)$$

In our experiment we trained three models in all, the original one(fig.3), the one used sparse train(fig.4) and the pruned one(fig.5). For each model, we collected how their loss and performance changed by each epoch and is shown below.

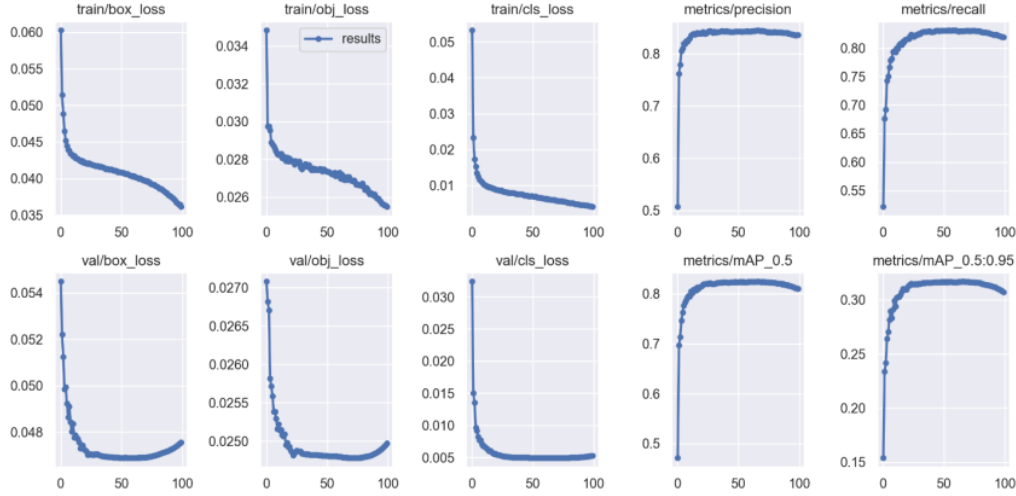


Figure 3 Results of original model

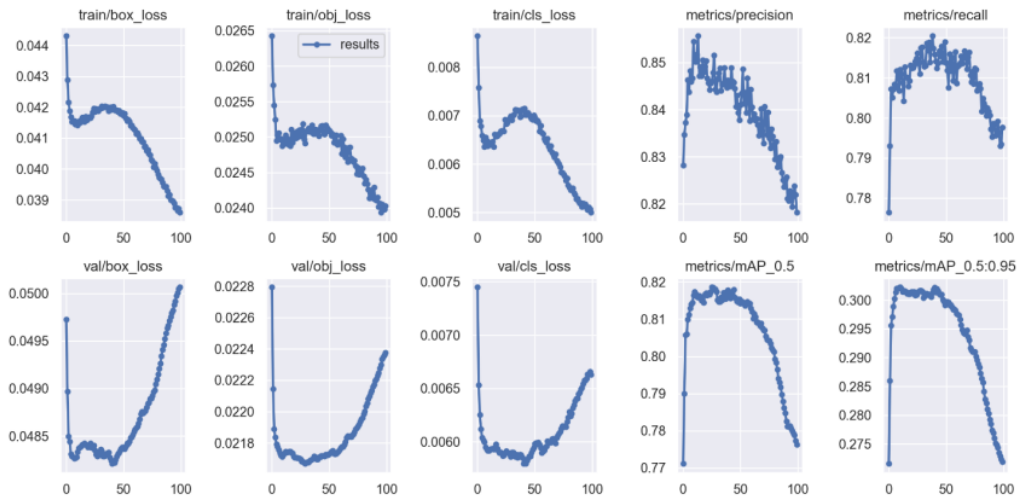


Figure 4 Results of model after sparse training

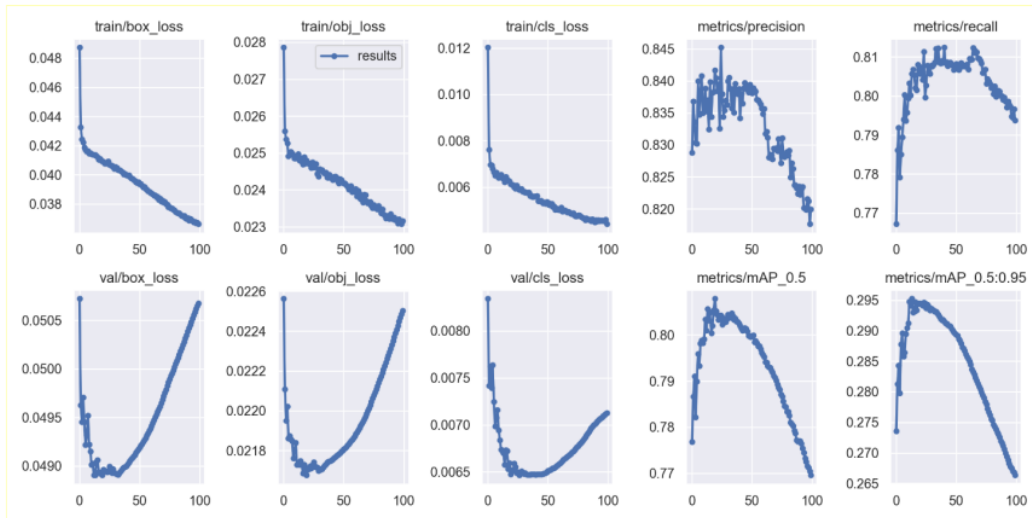


Figure 5 Results of pruned model

First, we used YOLOv5 directly to train the model. From fig.1, we found that the loss on training set decrease rapidly at the beginning epochs of training and maintain a slower decreasing trend in later epochs. Also, Precision and Recall reached their peak at no longer than 25 epochs. It shows that YOLOv5 is capable of dealing with digits recognition problems. The training time of each epoch in YOLOv5 took about 12 minutes on a GeForce RTX 2070s and the reasoning time for each image was about 2.8ms which had an FPS at 357/s. The mAP@0.5 is about 0.82.

Secondly, we implemented sparse training on YOLOv5. From fig.2, we found that the loss kept decreasing until epoch 15 and had a reverse. After about 20 epochs of slow increase, there appeared another reverse and the loss on training set began to decrease again. This was because the training was finished at about epoch 15 and the later epochs led to a overfitting. The training time of each epoch in YOLOv5 took about 14 minutes on a GeForce RTX 2070s and the reasoning time for each image was about 3.0ms which had an FPS at 333/s. The mAP@0.5 is about 0.81.

Lastly, we pruned the sparse trained model and implemented the result on YOLOv5. From fig.3, we found that the model had a higher speed of convergence. Precision and Recall reached peak at about epoch 25. The training time of

each epoch in Yolov5 took about 8 minutes on a GeForce RTX 2070s and the reasoning time for each image was about 2.6ms which had an FPS at 384/s. It showed that the pruned model required a much less training time than the original Yolov5 while it remained almost the same performance as the original one. Also, the pruned model had an increase in the speed of reasoning. The mAP@0.5 is about 0.80.

In summary, by using efficient training techniques like L1 regularization and channel pruning, we successfully optimized the Yolov5 model to reduce its size, speed up training, and increase inference speed. Our experiments show that the optimized Yolov5 model is 9% smaller than the original, with only a 2 percent drop in mAP. At the same time, training speed increased by 33% and inference speed increased by 7%. It's clear that these sparse training methods achieved great results in optimizing deep learning models.



Figure 6 Labels of street view pictures with house numbers



Figure 7 Prediction of street view house numbers

7. CONCLUSION

We implemented sparse training and pruning on Yolov5 to help improve model performance in recognizing digits in real world environments. With our experiments, the use of sparse training and pruning remarkably reduce the training time required and speed up reasoning. Also it takes a smaller space to store the model while has little loss in its performance. Through our approach, we improved Yolov5 and successfully implement Yolov5 in real world digits recognition.

AUTHORS' CONTRIBUTIONS (字体 Times New Roman, 字号 11.5, 加粗, 段前 12 磅, 段后 8 磅)

The title "AUTHORS' CONTRIBUTIONS" should be in all caps.

ACKNOWLEDGMENTS (字体 Times New Roman, 字号 11.5, 加粗, 段前 12 磅, 段后 8 磅)

The title "ACKNOWLEDGMENTS" should be in all caps and should be placed above the references. The references should be consistent within the article and follow the same style. List all the references with full details.

REFERENCES

- [1] 王泰刚.基于神经网络的手写字符识别系统.2013.大连海事大学,MA thesis.
- [2] David G. Lowe."Distinctive Image Features from Scale-Invariant Keypoints.." International Journal of Computer Vision 60.2(2004). doi:10.1023/B:VISI.0000029664.99615.94.
- [3] Dalal N., Triggs B. "Histograms of oriented gradients for human detection". 2005 IEEE Conference on Computer Vision and Pattern Recognition.IEEE,2005
- [4] Kussul,Emst, and Tatiana Baidyk. "Improved method of handwritten digit recognition tested on MNIST database." Image and Vision Computing 22.12 (2004): 971-981.
- [5] Chen, Li, et al. "Beyond human recognition: A CNN-based framework for handwritten character recognition." 2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR). IEEE, 2015.
- [6] Redmon, Joseph, et al. "You only look once: Unified, real-time object detection." Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.
- [7] Huang, Rachel, Jonathan Pedoeem, and Cuixian Chen. "YOLO-LITE: a real-time object detection algorithm optimized for non-GPU computers." 2018 IEEE international conference on big data (big data). IEEE, 2018.
- [8] Cao, Langcai, et al. "A text detection algorithm for image of student exercises based on CTPN and enhanced YOLOv3." IEEE Access 8 (2020): 176924-176934.
- [9] Guo Qiang, et al. "Hybrid CNN-HMM Model for Street View House Number Recognition." Asian Conference on Computer Vision.2014.
- [10] Hochuli, Andre G., et al. "Handwritten digit segmentation: Is it still necessary?." Pattern Recognition 78 (2018): 1-11.
- [11] 史素霞, 常婉秋, and 宋志英."基于 UCI 数据集的 OCR 光学字符识别." 科技创新与应用 12.35(2022):50-53. doi:10.19981/j.CN23-1581/G3.2022.35.012.

YOLO Sparse Training and Model Pruning for Street View House Numbers Recognition

ORIGINALITY REPORT

18%

SIMILARITY INDEX

10%

INTERNET SOURCES

17%

PUBLICATIONS

7%

STUDENT PAPERS

PRIMARY SOURCES

1	acec.event.upi.edu Internet Source	1 %
2	"Advances in Artificial Intelligence and Security", Springer Science and Business Media LLC, 2022 Publication	1 %
3	www.datasetlist.com Internet Source	1 %
4	"Advanced Computer Architecture", Springer Science and Business Media LLC, 2020 Publication	1 %
5	Submitted to University of Sussex Student Paper	1 %
6	Fuxiang Liu, Song He, Zhisheng Chen. "Improved Alexnet-Based Character Detection Method", Journal of Physics: Conference Series, 2022 Publication	1 %
7	ntnuopen.ntnu.no Internet Source	

1 %

8

www.mdpi.com

Internet Source

1 %

9

Zhao Zekuan, He Chunlin. "Research on Defect Detection Method of Drainage Pipe Network Based on Deep Learning", 2022 19th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP), 2022

Publication

1 %

10

Submitted to University College London

Student Paper

<1 %

11

Xuan Yang, Shanshan Li, Zhengchao Chen, Jocelyn Chanussot, Xiuping Jia, Bing Zhang, Baipeng Li, Pan Chen. "An attention-fused network for semantic segmentation of very-high-resolution remote sensing imagery", ISPRS Journal of Photogrammetry and Remote Sensing, 2021

Publication

<1 %

12

Submitted to University of Melbourne

Student Paper

<1 %

13

caai.cn:8080

Internet Source

<1 %

14

www.frontiersin.org

Internet Source

<1 %

15

Submitted to University of Technology,
Sydney

Student Paper

<1 %

16

Submitted to University of Wollongong

Student Paper

<1 %

17

downloads.hindawi.com

Internet Source

<1 %

18

"Intelligent Computing Theories and
Application", Springer Science and Business
Media LLC, 2016

Publication

<1 %

19

"Intelligent Equipment, Robots, and Vehicles",
Springer Science and Business Media LLC,
2021

Publication

<1 %

20

Jia Zhao. "Design of Intelligent Guidance
Algorithm for Language Training in Cross-
cultural Communication based on Multicast
Multimedia Fusion SSD Framework", 2022 7th
International Conference on Communication
and Electronics Systems (ICCES), 2022

Publication

<1 %

21

Sirui Shen, Wenqing Song, Xinyu Wang, Xinyu
Shao, Yuxiang Fu, Zhonghai Lu, Li Li. "A
Hierarchical Parallel Discrete Gaussian

<1 %

Sampler for Lattice-Based Cryptography",
2022 IEEE International Symposium on
Circuits and Systems (ISCAS), 2022

Publication

22

Xingyuan Wang, Xuan Chen. "Image encryption algorithm based on cross-scrambling and rapid-mode diffusion", The Visual Computer, 2022

Publication

<1 %

23

scholar.dkyobobook.co.kr

Internet Source

<1 %

24

www.hindawi.com

Internet Source

<1 %

25

Zhen Zhang, Huiying Chen, Ruijie Xiao, Qingjie Li. "Research on Smoking Detection Based on Deep Learning", Journal of Physics: Conference Series, 2021

Publication

<1 %

26

docplayer.net

Internet Source

<1 %

27

ufldl.stanford.edu

Internet Source

<1 %

28

Advances in Intelligent Systems and Computing, 2015.

Publication

<1 %

29

A.G. Hochuli, L.S. Oliveira, A.S. Britto Jr, R. Sabourin. "Handwritten digit segmentation: Is it still necessary?", Pattern Recognition, 2018

Publication

<1 %

30

Lichao Liu, Jing Liang, Jianqing Wang, Peiyu Hu, Ling Wan, Quan Zheng. "An improved YOLOv5-based approach to soybean phenotype information perception", Computers and Electrical Engineering, 2023

Publication

<1 %

31

Lin Zhang, Yaqi Sun, Wenli Chen, Xiaoman Liang. "Chapter 24 Traffic Sign Recognition System Based on YOLOv5", Springer Science and Business Media LLC, 2022

Publication

<1 %

32

Qingjin Xu, Rui Fu, Fuwei Wu, Biyao Wang. "Roadside estimation of a vehicle's center of gravity height based on an improved single-stage detection algorithm and regression prediction technology", IEEE Sensors Journal, 2021

Publication

<1 %

33

Wenjing Hong, Guiying Li, Shengcai Liu, Peng Yang, Ke Tang. "Multi-objective evolutionary optimization for hardware-aware neural network pruning", Fundamental Research, 2022

Publication

<1 %

34

Yunhai Song, Zhenzhen Zhou, Qiang Li, Yizhou Chen, Pengfei Xiang, Qian Yu, Liang Zhang, Yanfeng Lu. "Intrusion detection of foreign objects in high-voltage lines based on YOLOv4", 2021 6th International Conference on Intelligent Computing and Signal Processing (ICSP), 2021

Publication

<1 %

35

Handbook of Document Image Processing and Recognition, 2014.

Publication

<1 %

36

[programs.wiki](https://www.programs.wiki)

Internet Source

<1 %

37

"Computer Vision - ACCV 2014 Workshops", Springer Nature, 2015

Publication

<1 %

38

"MultiMedia Modeling", Springer Science and Business Media LLC, 2018

Publication

<1 %

39

"Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications", Springer Science and Business Media LLC, 2018

Publication

<1 %

40

Haoxiang Liang, Huansheng Song, Xu Yun, Shijie Sun, Yingxuan Wang, Zhaoyang Zhang. "Traffic incident detection based on a global

<1 %

41

Fan Yang, Deming Yang, Zhiming He, Yuanhua Fu, Kui Jiang. "Automobile Fine-Grained Detection Algorithm Based on Multi-Improved YOLOv3 in Smart Streetlights", Algorithms, 2020

Publication

<1 %

42

Jie Li, Lei Huang, Zhiqiang Wei, Wenfeng Zhang, Qibing Qin. "Multi-task learning with deformable convolution", Journal of Visual Communication and Image Representation, 2021

Publication

<1 %

43

Shun Luo, Juan Yu, Yunjiang Xi, Xiao Liao. "Aircraft Target Detection in Remote Sensing Images Based on Improved YOLOv5", IEEE Access, 2022

Publication

<1 %

Exclude quotes Off

Exclude matches Off

Exclude bibliography On