Transactions on Computer Science and Intelligent Systems Research
ISSN: 2960-1800, eISSN: 2960-2238 | Volume 5
AIDML 2024

Detection of road defects with weak small samples based on multiple deep learning models

Huaizheng Lu [1, *], Xinyi Wu [2], Dedong Zhang [1]

[1] College of Computer Engineering, Jimei University, Xiamen, China, 361021

[2] School of Ocean Information Engineering, Jimei University, Xiamen, China, 361021

* Corresponding author: luhuaizheng@jmu.edu.cn

Abstract. Potholes on roads not only jeopardize traffic safety and driving comfort but also necessitate efficient detection and maintenance measures. Traditional manual detection methods are labor-intensive and time-consuming. Thus, developing automated solutions is imperative. This study addresses this challenge by constructing a dataset and evaluating various deep learning models, including GoogleNet, VGG16, ResNet50, AlexNet, YOLOV5, and YOLOV8, for pothole detection. YOLOV8 emerges as the optimal choice due to its superior accuracy. However, accurately estimating pothole areas proves challenging due to their irregular shapes. To mitigate this, an innovative algorithm is proposed, integrating YOLO with a pre-trained segmentation model. This enables precise pixel-level delineation of pothole areas. Additionally, the algorithm incorporates the intersection over union (IOU) metric to calculate the ratio of pothole area to the total image area. By enhancing both detection and area estimation, this approach holds promise for improving road safety and facilitating maintenance efforts. Automated detection and accurate area estimation not only save time and resources but also provide crucial data for prioritizing and planning road repair and maintenance tasks.

Keywords: Road Defects Detection; YOLO; Segment Anything.

1. Introduction

Potholes have become a common problem in urban traffic in recent years, potholes on roads not only cause inconvenience to motorists but can also lead to vehicle damage, accidents and traffic congestion. Therefore, it becomes critical to ac- curately and efficiently detect and repair road potholes. With the continuous advancement of technology, road pothole detec- tion technology has gradually developed. Traditional road pot- hole detection mainly relies on manual inspection. This method is not only time-consuming and labor-intensive, but also prone to subjectivity and omissions. However, with the rapid devel- opment of computer vision and machine learning, automated road pothole detection technology has become possible. These technologies use cameras, radar or other sensor devices [1], combined with image processing and machine learning algorithms [2, 3], to accurately detect potholes on the road in real time.

One of the key challenges in road pothole detection is to ac- curately extract pothole information from large amounts of sen- sor data. This requires powerful computing power and efficient data processing algorithms. In addition, since road potholes come in various shapes and sizes, the algorithm needs to have certain robustness and adaptability [4]. In addition, the timeli- ness and accuracy of detection results are also issues that need to be solved with road technology [5].

This article aims to compare the detection effects of pothole defects using traditional deep learning models and the popular YOLO series detection models [6]. At the same time, we hope to solve practical problems, that is, in some application scenar- ios, it is necessary to accurately calculate the proportion of the pothole area to the entire image area, and to accurately segment the pothole graphics.

2. Related Works

2.1. Road Defect Detection

There have been many studies on road defect detection in recent years, which not only achieve lightweight deployment but also continuously improve the accuracy of the algorithm. Aure´lien et al. [7] proposed a method to automatically distin- guish between defective and non-defective pavement images, showing fluctuations on small scales and a certain uniformity on larger scales, which can be applied by performing super- vised learning based on AdaBoost. All types of defects present in these images. Wang et al. proposed an enhanced road de- fect detection algorithm called BL-YOLOV8[8] , which im- proved the speed of the model, expanded the receptive field of the model and improved the accuracy of target detection by in- troducing the SimSPPF module to optimize the feature pyramid layer, and confirmed that the improved YOLOV8 algorithm can effectively identify road defects. Chatterjee et al. proposed a method based on machine learning [9]. They first used the su- perpixel method to process the road image into smaller coherent image areas, and then used a classification model to divide these areas into cracked areas and non-cracked areas. Finally, it was confirmed that it was successful. Different road surface con- ditions and crack types are effectively handled while locating defective areas in scene images.

2.2 Defect Detection

In road defect detection, we here are more concerned about the detection of potholes. Strazdins et al. proposed a mobile sensing system for road pothole detection using smartphones based on Android operating system [10], and briefly discussed efficient algorithms that can be applied to road pothole detec- tion. Pereira et al. proposed a low-cost solution for detect- ing road pothole images using a convolutional neural network (CNN), and achieved extremely high accuracy under test con- ditions using 500 images [11]. Dhiman et al. briefly summa- rized various identification strategies for road potholes, devel- oped and studied two technologies based on stereoscopic vision analysis of the road environment in front of the vehicle, and designed two pothole detection models based on deep learning [12].

3. Background

3.1. GoogleNet

The GoogleNet model is a 22-layer deep convolutional neu- ral network [13]. Its core feature is the introduction of the In- ception module structure. Its core idea is how to use convolu- tion kernels of different sizes (such as$1 \times 1, 3 \times 3, 5 \times 5$, etc.) and combining the outputs allows the same layer in the network to capture information at different scales. At the same time, compared with the traditional CNN network, this network uses maximum pooling to connect all feature maps and output them. And because it uses global average pooling instead of fully con- nected layers, it greatly reduces the number of parameters of the model. We have fine-tuned the model and used 1×1 convolu- tion to reduce calculations, so its computational efficiency is still very high. In addition, we introduced an auxiliary classi- fier and designed a small auxiliary network structure on some intermediate layers in the network to help with classification, thereby regularizing the overall network and avoiding gradient disappearance.

3.2. Visual Geometry Group16(VGG16)

Visual Geometry Group (VGG) mainly uses 3×3 convolu- tion kernels and 2×2 pooling layers [14]. It replaces large con- volution kernels by continuously using multiple small convo- lution kernels. The architecture is simple and regular, easy to implement, and efficient at the same time. The convolution uses 3×3 convolution kernels multiple times, allowing the network to reach a certain depth while maintaining computational effi- ciency. The network structure is as shown in Figure 1.

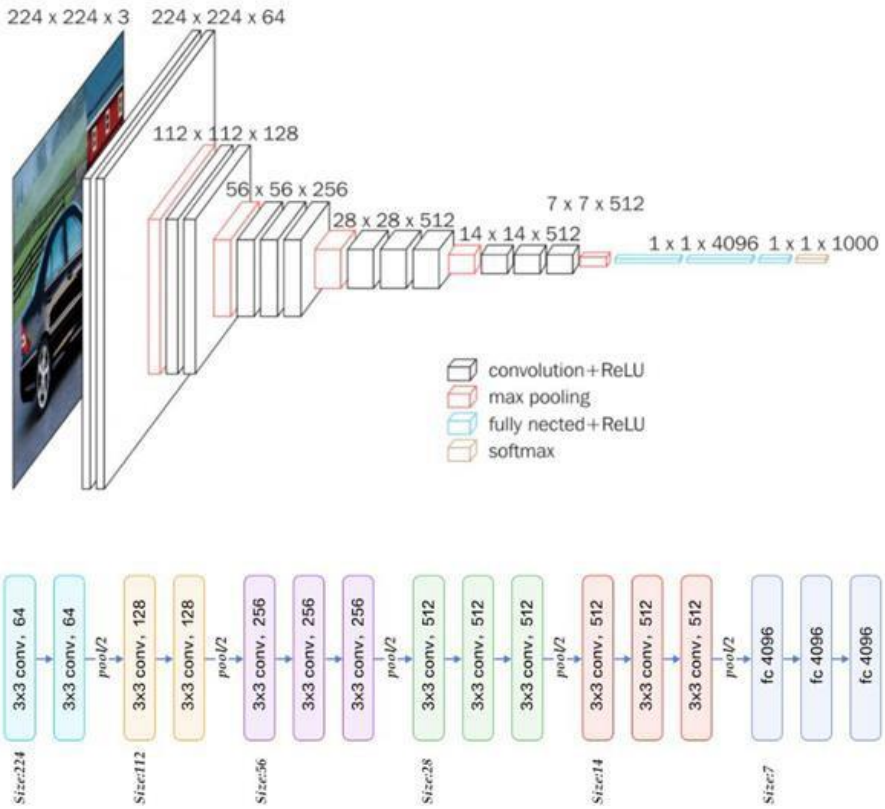

Figure 1. VGG16 network framework

A 3×3convolution kernel has 9 weight parameters, while a 5×5 convolution kernel requires 25 weight parameters. There- fore, using a 3×3 convolution kernel can greatly reduce the number of parameters in the network, thereby reducing the risk of overfitting. Multiple 3×3convolution kernels can be con- nected in series to form a convolution kernel with a larger re- ceptive field, and this combination has stronger nonlinear capa- bilities. In VGG, using 3×3 convolution kernel multiple times is equivalent to using a larger convolution kernel, which can improve the feature extraction capability of the network. It is very suitable for the small sample environment used to solve this issue.

3.3. AlexNet

AlexNet is a convolutional neural network suitable for image recognition. Compared with the LeNet network structure that appeared before this, it has a more complex network structure and more parameters. Its overall network structure is relatively simple, including only 1 input layer, 5 convolutional layers, 2 fully connected layers and an output layer. The structure dia- gram is as follows in Figure 2.

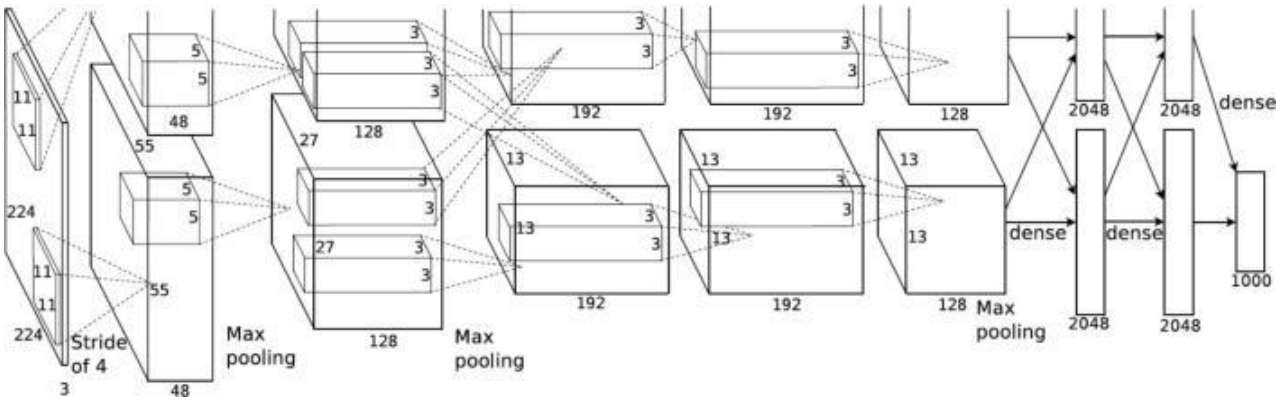

Figure 2. AlexNet network framework

Suppose we have an input image X, a convolution kernel W and a bias b. The output Y of the convolutional layer can be expressed as:

$$Y_{i,j} = \sum_m \sum_n X_{i+m,j+n} W_{m,n} + b \tag{1}$$

This network design roughly captures an important property of natural images, that is, object identity is invariant to changes in lighting intensity and color, and is therefore suitable for use in feature extraction and recognition classification for road pot- hole detection.

3.4. Segment Anything (SAM)

SAM adopts an encoder-decoder structure like U-Net, in which the encoder part consists of multiple convolutional layers and pooling layers to extract image features; the decoder part consists of multiple deconvolution layers. layer and an up-sampling layer, used to restore the feature map to the original image size and generate segmentation results. At the same time, SAM uses a multi-task loss function based on cross-entropy, which includes pixel-level classification loss and bounding box-level regression loss. [15] The classification loss is used to measure which category each pixel belongs to (such as foreground or background), and the regression loss is used to adjust the bounding box position of each pixel to better match the object. To improve the robustness and generalization ability of the model, SAM uses a variety of data enhancement techniques, such as random rotation, scaling, cropping, flipping, etc., as well as color space transformation and noise addition. To speed up model training and improve segmentation accuracy, SAM usually uses pre-trained image classification models (such as ResNet, VGG, etc.) as the initial weights of the encoder to better extract image features. The network structure is as shown in Figure 3.

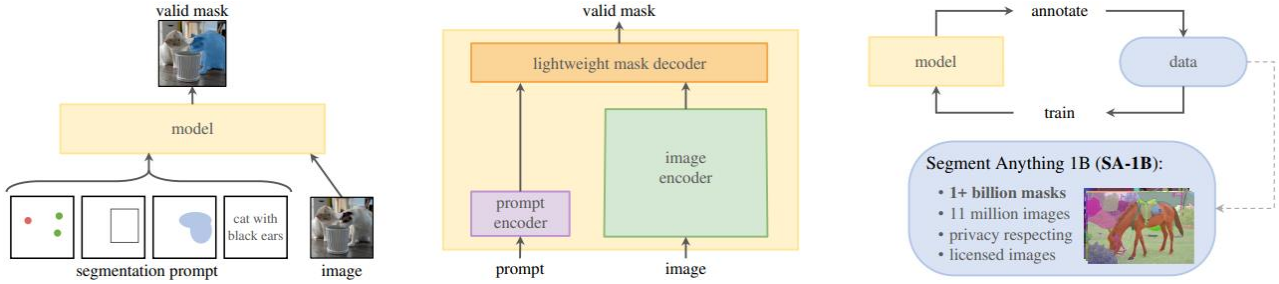

Figure 3. SAM network framework

4. Experiment

4.1. Performance evaluation of different four traditional CNN models

To measure the quality of a machine learning model, the model needs to be used to predict the samples in the test set, and the evaluation score is calculated based on the prediction results. For classification problems, the evaluation criteria we use include accuracy, precision, recall, and F-value. We assume that the given test set $\tau = \{(x^{(1)}, y^{(1)}), \ldots, (x^{(n)}, y^{(n)})\}$, assume label $y^{(n)} \in \{1, \ldots, C\}$, use the learned model $f(x; \theta^*)$ to predict each sample in the test set,the result is $\{\hat{y}^{(1)}, \ldots, \hat{y}^{(n)}\}$. The accuracy A can be expressed as:

$$A = \frac{1}{N} \sum_1^N \tau(y^{(n)} = \hat{y}^{(n)}) \tag{2}$$

The recall rate refers to the recall rate of category c, which is the proportion of correct predictions among all samples whose true label is category c:

$$R_c = \frac{TP_c}{TP_c + FN_c} \tag{3}$$

F Measure is a comprehensive indicator, which is the harmonic average of precision and recall.It can be calculated as:

$$F_c = \frac{(1+\beta^2) \times P_c \times R_c}{\beta^2 \times P_c + R_c} \tag{4}$$

Combining the evaluation indicators mentioned above, we can calculate the detection performance of GoogleNet, VGG16, ResNet50 and AlexNet respectively. The results are as shown in Table 1.

Table 1. Performance comparison between different models

Model name	Accuracy	Recall	Consistency	F-value	Time cost
GoogleNet	0.9314	0.9707	0.9104	0.9892	4s/step
ResNet50	0.6287	0.6831	0.6831	0.7292	14s/step
VGG16	0.9308	0.8534	0.8980	0.9474	12s/step
AlexNet	0.2251	0.8285	0.7541	0.7952	8s/step

The GoogleNet model performs best in terms of accuracy, recall, precision and F-score, with the highest accuracy and re- call and the lowest time overhead. The VGG16 model performs better in accuracy and precision, but has a slightly lower recall rate. The ResNet50 model performs generally in various indica- tors, while the AlexNet model has lower accuracy, but relatively higher recall and precision. Therefore, by evaluating the model in different dimensions, this article believes that GoogleNet has better performance in dealing with pothole road detection and identification problems.

4.2. Performance evaluation of YOLO models

The YOLOV5 model also considers the target confidence loss, which helps to better distinguish pothole areas and non-pothole areas during the detection process. The parameters of the YOLOV5 pre-trained model are shown in Table 2.

Table 2. YOLOV5 pre-trained model parameters

Model	Size (pixels)	mAPval	Speed (ms)	FLOPS (B)
YOLOV5s	640	56.8	98	16.5
YOLOV5m	640	64.1	224	49.0
YOLOV5l	640	67.3	430	109.1
YOLOV5x	640	68.9	766	205.7

All the above checkpoints are calculated after the model has been trained for 300 epochs. If our computing power is sufficient, we can analyze it: the input pixel size of the four basic models of YOLOV5 is 640, in size There is no difference between the several models in this respect. At the same time, what we need is to improve the detection accuracy. The detection speed can be given a lower priority, so we only need to consider the mean average precision (mAP) of the model. It is not difficult to observe that the mAP and FLOPS values of YOLOV5x are significantly better than other models, so the YOLOV5x pre-trained model was selected for training.

The YOLOV8 model is the latest version of the YOLO series model. It adopts an Anchor-Free detection method, which has higher detection accuracy and faster detection speed than tra- ditional Anchor-based detection. Overall, YOLOV8 and YOLOV5 are basically the same. The optimization part is that in the YOLOV8-Head part, Cls and Box are predicted sepa- rately, and Anchor-Based is replaced by Anchor-Free, and C2f is used in Backbone and Neck of YOLOV8. structure. We ana- lyze

multiple pre-trained models of the YOLOV8 series pro- vided (such as YOLOV8s, YOLOV8m, etc.), the results are shown in Table 3.

Table 3. YOLOV8 pre-trained model parameters

Model	Size (pixels)	mAPval	Speed (ms)	FLOPS (B)
YOLOV8s	640	37.3	80.4	8.7
YOLOV8m	640	44.9	128.4	28.6
YOLOV8l	640	50.2	234.7	78.9
YOLOV8x	640	52.9	375.2	165.2

Through the analysis of the parameters of the YOLOV8 pre- training model, several groups of models are similar in terms of accuracy and performance. The main factors affecting the model effect are the application scenarios and scale of the train- ing set. Therefore, the optimal model is selected by training all YOLOV8 pre-trained models and comparing the training re- sults.The model training visualization effects of YOLOV5x and YOLOV8 are as shown in Figure 4 and Figure 5 to Figure 9.

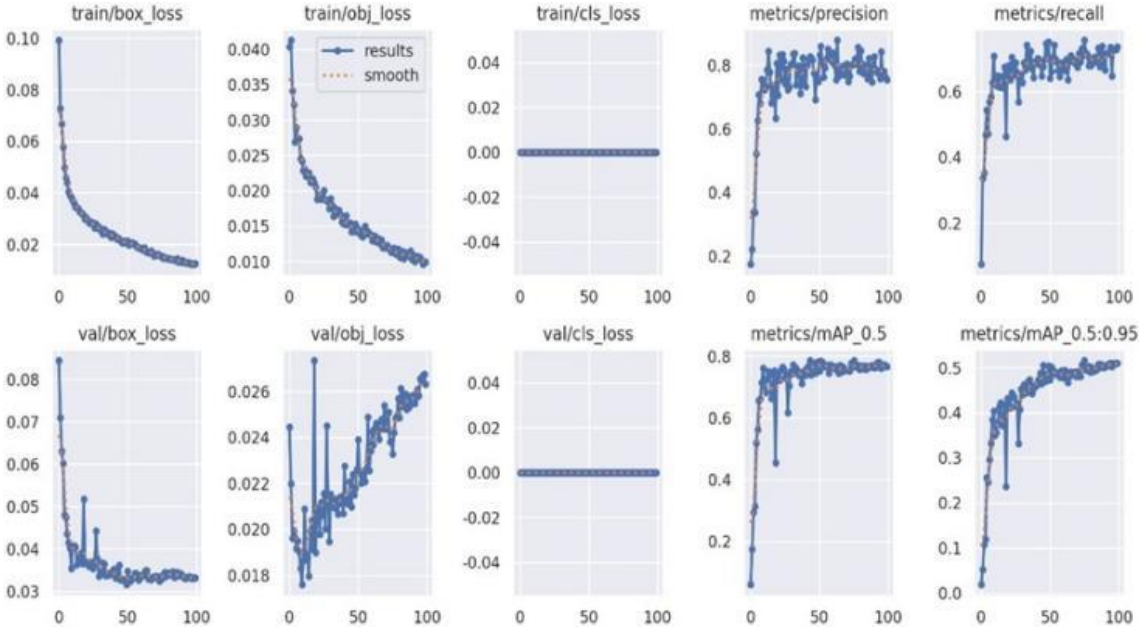

Figure 4. YOLOV5x training visualization

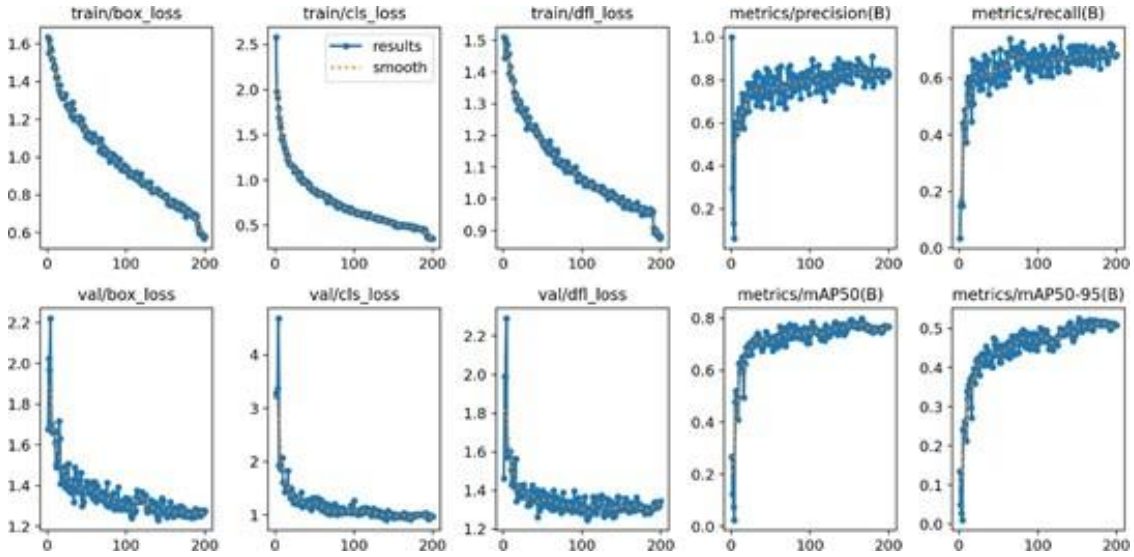

Figure 5. YOLOV8n training visualization

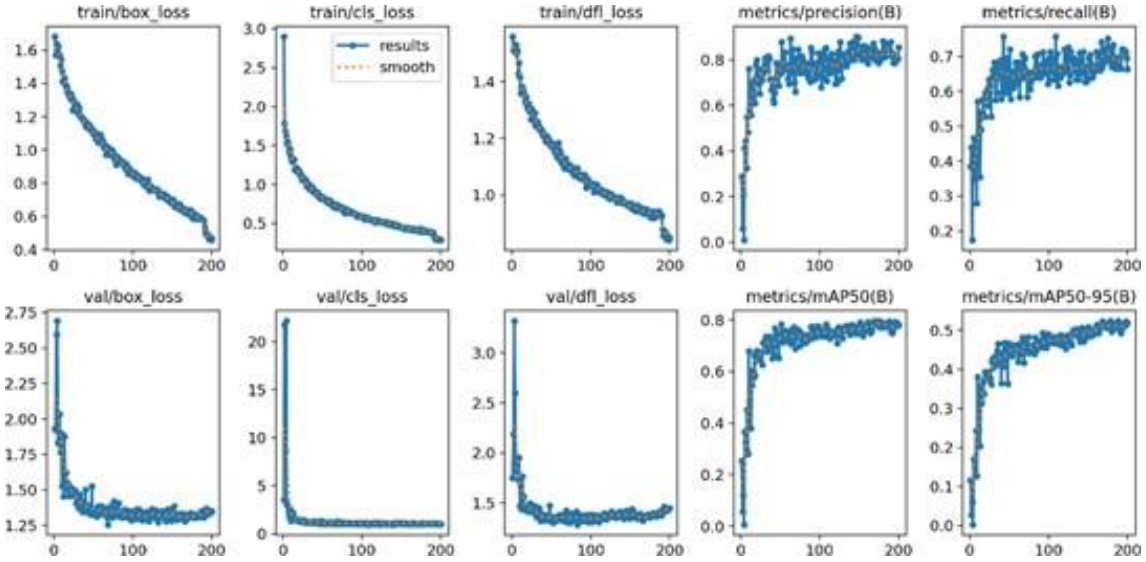

Figure 6. YOLOV8s training visualization

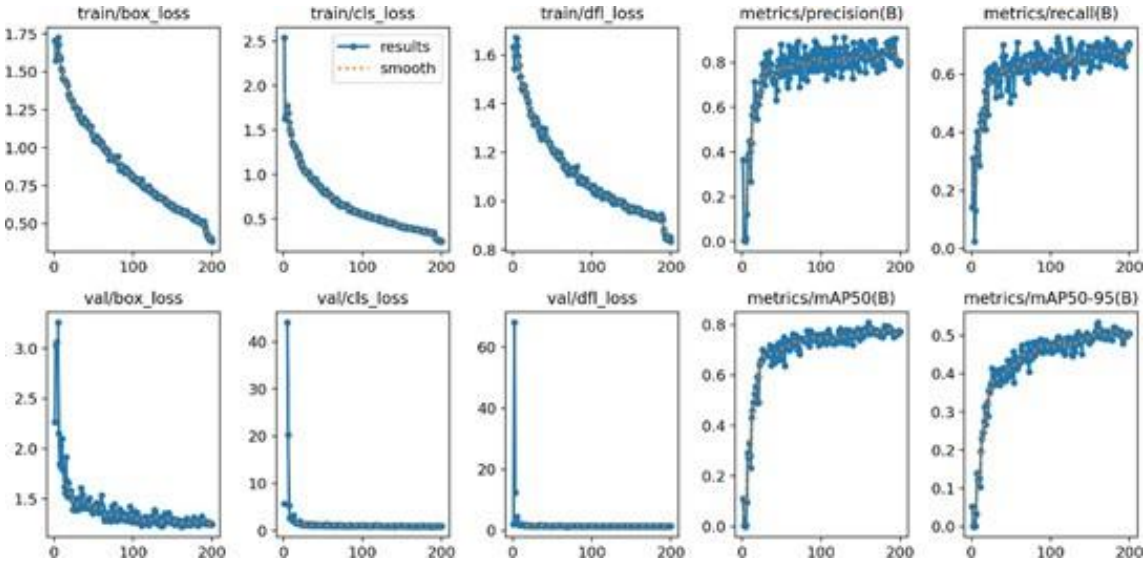

Figure 7. YOLOV8m training visualization

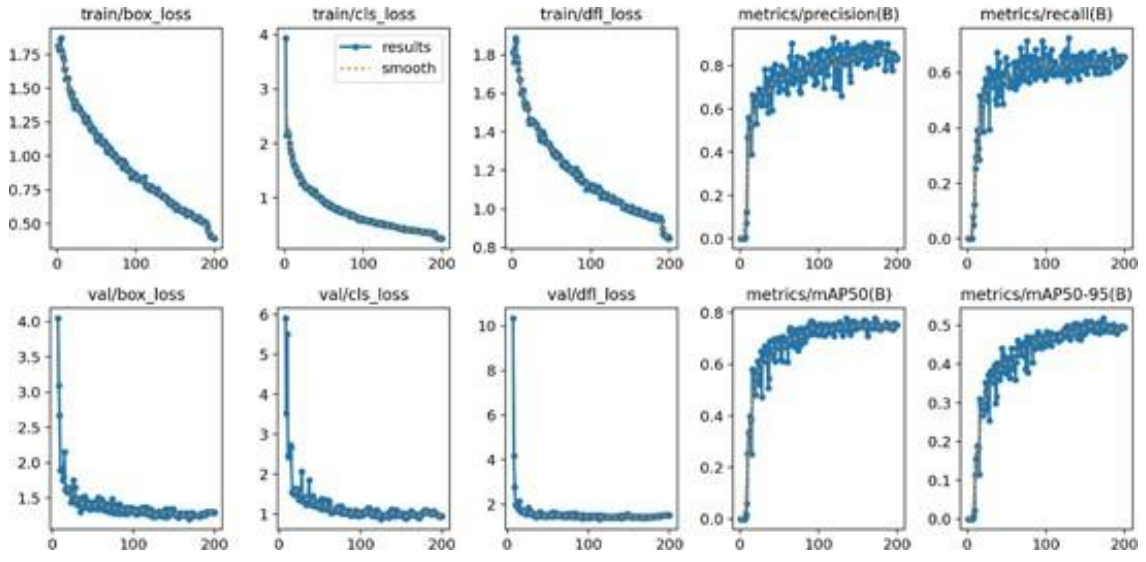

Figure 8. YOLOV8l training visualization

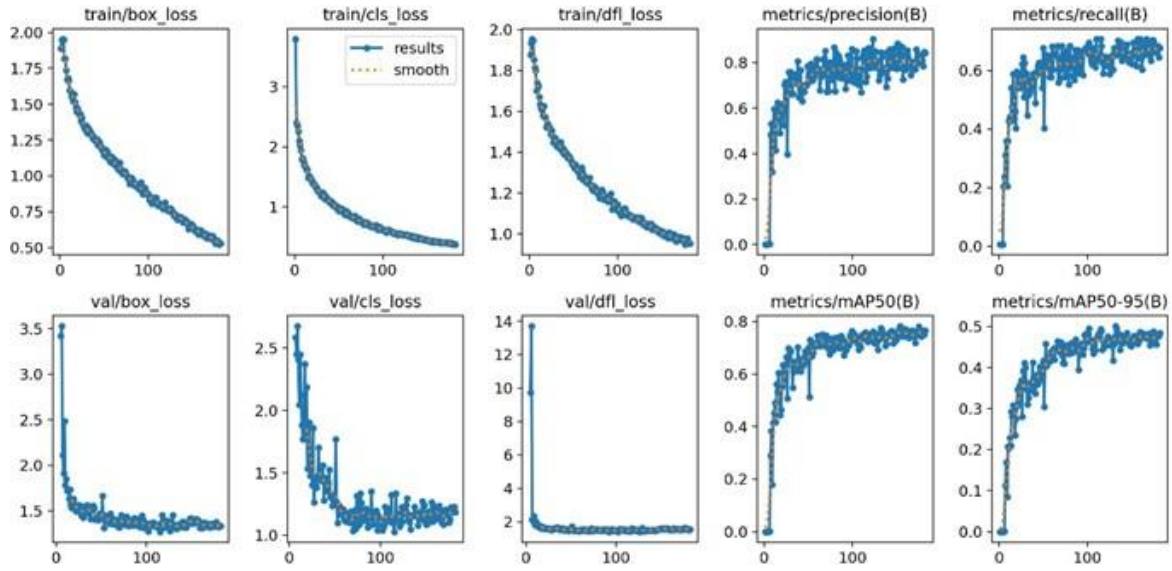

Figure 9. YOLOV8x training visualization

By comparing the visualization of the training process, we can clearly see that the val/cls loss of the YOLOV5 model, YOLOV8n, YOLOV8l, and YOLOV8x model fluctu- ates greatly during the training process, which means that the model's ability to distinguish features still has room for im- provement, and YOLOV8s, the loss of the YOLOV8m model drops to a smooth curve, and it can be considered that the results of these two models are better.

Then we use pix2pixGAN to generate an adversarial network to convert visible light images as input into the desired" pseudo-infrared" effect. The dataset used to train pix2pixGAN is the KAIST multispectral pedestrian dataset. Our experimental results show that pix2pixGAN can generate infrared videos of different qualities for various visible light videos in the KAIST dataset. At the same time, we still use the feature extraction method mentioned before, that is, using the Kinectics- 400 visible light data set to pretrain the I3D model (pretrain) and using the InfAR data set to retrain the I3D model (retrain). We believe that a universal video quality assessment method based on FVD should be applicable to various infrared videos (such as InfAR from human actions and KAIST from pedestri- ans). Therefore, the dataset of the model used to extract video features (e.g., InfAR dataset) is not consistent with the dataset of the machine-generated videos tested (e.g. KAIST dataset) in this paper.

To choose a better model on YOLOV8s and YOLOV8m, it is difficult to do so from the parameter changes during the training process, so we choose to manually annotate by selecting some test images, and calculate the consistency of the manual annotation and model annotation results. to deter- mine which model has the best effect. We select 100 images and through manual annotation, we can get the number of two types of data with potholes and without potholes, which is 78 images with potholes and 22 images without potholes. By calculating the results of the two models on the images with potholes the consistency of the results can be obtained by counting the per- centage of the number and the manual annotation results. The higher the consistency, the higher the accuracy of the model. We believe that the accuracy is higher. The model comparison results of YOLOV8s and YOLOV8m is Table 4.

Table 4. Small sample test results of YOLOV8s and YOLOV8m

Model	With potholes	Without potholes	consistency
YOLOV8s	63	37	80.77%
YOLOV8m	59	41	75.64%

The optimal training model YOLOV8s has a training accu- racy of 85.4%. However, the final output result of the model is to use a rectangular frame to frame the potholes, and the text result obtained is

also used to describe the location of the frame selection. If the area proportion of the second question is cal- culated based on this, the non-selected part of the frame will be the pothole area may cause a large loss of accuracy and lead to a large deviation in the result. Therefore, this article mi- grates SAM to our frame selection result data set, and performs secondary pothole segmentation on the content inside the frame selection. The effect is shown in Figure 9.

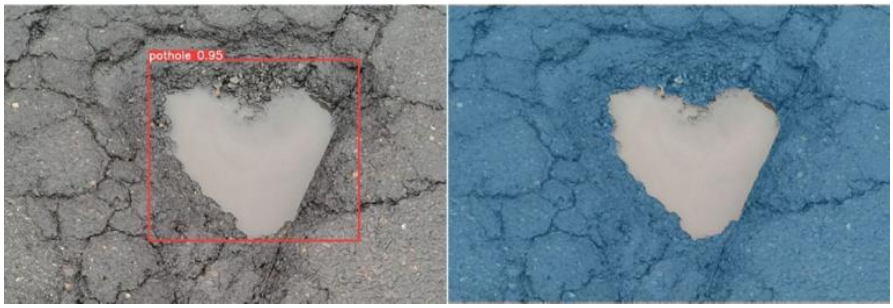

Figure 9. Secondary segmentation result of SAM

Because of this secondary segmentation, the calculation formula for the proportion of pothole area in the image can be obtained as:

$$R_{pothole} = \frac{S_{SAM\_pix}}{S_{All\_pix}} \tag{4}$$

Where S_{SAM_pix} represents the sum of the pixel areas of the potholes segmented by SAM in the image, and S_{All_pix} represents the total pixel area of the image. Add constraints based on this calculation: (1). Multiple calculations will not be performed to detect partial overlapping areas. Since SAM mask segmentation may separate two connected potholes, the area of the intersection will be calculated twice, so add the condition of IOU=0 when calculating the area ratio, otherwise the potholes will not be separated. calculation, thereby ensuring that no area portion is double-calculated. (2). The area ratio of the detection part may exceed 100%. Since the detection accuracy of YOLOV8 does not reach 100%, false detection effects may occur when the image is blurred. In this case, it is necessary to constrain the range of the area ratio of the detection part. Therefore, we add a constraint: the pothole area proportion interval is [0, 0.95]. When the proportion is 0, it means that no potholes are detected currently.

5. Conclusion

This article compares four traditional CNN networks: GoogleNet, VGG16, ResNet50, and AlexNet. The optimal model is GoogleNet. However, training traditional CNN models requires a large amount of labeled data, and since deep models have many parameters, insufficient training or overtraining will also lead to model underfitting or overfitting. At the same time, these traditional CNN models may not be able to meet the requirements on some devices. Real-time processing requirements. Pothole detection places higher demands on the entire model and equipment, requiring an understanding of the contextual information of the entire scene, and traditional CNN may be somewhat deficient in this regard. Therefore, we used the currently popular YOLO series target detection models and selected the YOLOV5 and YOLOV8 series models that are most suitable for pothole detection. Through the performance analysis of the pre-trained model on the test set, we agree that YOLOV8s can achieve the best results on the self-made data set of this article. And by comparing the identification results, the accuracy of the YOLO series models is much higher than the traditional CNN network. Therefore, based on the pothole detection map of YOLOV8s, this article proposes to use SAM to further segment the pothole content of the image. On this basis, the proportion of the pothole area to the entire image area can be further calculated. At the same time, our analysis the geometry of the potholes has a meaning.

In the future, this article believes that the test data set can be further expanded, and models with higher robustness and accuracy can be found through fine-tuning and model migration. At the same time, this article believes that using SAM to perform secondary segmentation on images also has other hidden values to be mined.

References

[1] Yao S, Guan R, Huang X, et al. Radar-camera fusion for object detection and semantic segmentation in autonomous driving: A comprehensive review [J]. IEEE Transactions on Intelligent Vehicles, 2023.

[2] Méndez M, Merayo M G, Núñez M. Machine learning algorithms to forecast air quality: a survey [J]. Artificial Intelligence Review, 2023, 56 (9): 10031 - 10066.

[3] Ali Y A, Awwad E M, Al-Razgan M, et al. Hyperparameter search for machine learning algorithms for optimizing the computational complexity [J]. Processes, 2023, 11 (2): 349. K. Elissa, “Title of paper if known,” unpublished.

[4] Song H, Huang S, Dong Y, et al. Robustness and generalizability of deepfake detection: A study with diffusion models [J]. arXiv preprint arXiv:2309.02218, 2023.

[5] Jakubec M, Lieskovská E, Bučko B, et al. Comparison of CNN-based models for pothole detection in real-world adverse conditions: overview and evaluation[J]. Applied Sciences, 2023, 13 (9): 5810.

[6] Wang Z, Cai Z, Wu Y. An improved YOLOX approach for low-light and small object detection: PPE on tunnel construction sites [J]. Journal of Computational Design and Engineering, 2023, 10 (3): 1158 - 1175.

[7] A. Cord, S. Chambon, Automatic Road defect detection by textural pattern recognition based on adaboost, Computer-Aided Civil and Infrastructure Engineering 27 (4) (2012) 244 – 259.

[8] X. Wang, H. Gao, Z. Jia, Z. Li, Bl-yolov8: An improved road defect detection model based on yolov8, Sensors 23 (20) (2023) 8361.

[9] S. Chatterjee, P. Saeedfar, S. Tofangchi, L. M. Kolbe, Intelligent Road maintenance: a machine learning approach for surface defect detection., in: ECIS, 2018, p. 194.

[10] A. Mednis, G. Strazdins, R. Zviedris, G. Kanonirs, L. Selavo, Real time pothole detection using android smartphones with accelerometers, in: 2011 International conference on distributed computing in sensor systems and workshops (DCOSS), IEEE, 2011, pp. 1 – 6.

[11] V. Pereira, S. Tamura, S. Hayamizu, H. Fukai, A deep learning-based approach for road pothole detection in timor leste, in: 2018 IEEE Interna- tional Conference on Service Operations and Logistics, and Informatics (SOLI), IEEE, 2018, pp. 279 – 284.

[12] A. Dhiman, R. Klette, Pothole detection using computer vision and learn- ing, IEEE Transactions on Intelligent Transportation Systems 21 (8) (2019) 3536 – 3550.

[13] M. Al-Qizwini, I. Barjasteh, H. Al-Qassab, H. Radha, Deep learning algo- rithm for autonomous driving using googlenet, in: 2017 IEEE intelligent vehicles symposium (IV), IEEE, 2017, pp. 89 – 96.

[14] A. Sengupta, Y. Ye, R. Wang, C. Liu, K. Roy, Going deeper in spik- ing neural networks: Vgg and residual architectures, Frontiers in neuro- science 13 (2019) 95.

[15] Kirillov A, Mintun E, Ravi N, et al. Segment anything [C]//Proceedings of the IEEE/CVF International Conference on Computer Vision. 2023: 4015 - 4026.