Technical University of Berlin

Fraunhofer Institute for Open
Communication Systems FOKUS

Faculty 4

Projekt Vertiefung / Vernetztes und
Automatisiertes Fahren
Winter term 2020/2021

Street Damage Detection and Rating

Team members:

Yilei Chen (451619)
Yiying Yang (414669)
Jie Chen (399118)

Supervised by:

Iskander Tschinibaew

March 7, 2023

Contents

1 Introduction

1.1 Background and Motivation

With the steady development of the economy, roads have also developed rapidly as a transportation link. The continuous expansion of the road traffic network not only brings convenience to people's travel, but also relieves the traffic congestion caused by the increase in the number of cars, but also brings great challenges to road maintenance. For example, during the use of the road surface, under the influence of load and natural factors for a long time, the road surface will suffer various damages and gradually lose its serviceability(As shown in Fig.1). Some damage problems even threaten traffic safety and pose a threat to people's life safety. Therefore, effective maintenance and maintenance of roads can not only improve the service life and operation management of roads but also reduce traffic accidents and improve road safety.

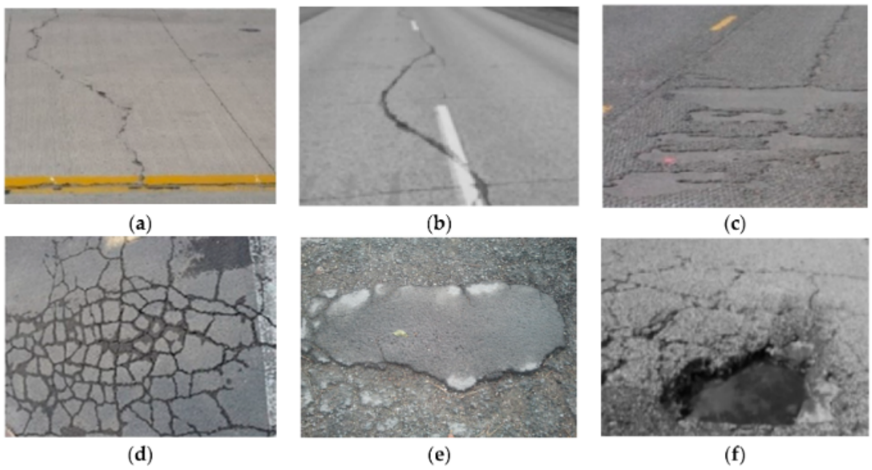


Figure 1: Types of road damage

Previously, crews relied primarily on the naked eye to see if damage existed on the road. However, with the development of image processing technology, the application of these techniques to pavement damage detection can both eliminate the subjective human element and automatically assess the condition of the pavement quickly and accurately.

1.2 Research Contents

In the second part of the introduction, the research process and the structure of the thesis are briefly summarised.

In the first step, the ZEB image data set was selected according to the shapefile. Six street shapefile files were provided. By analyzing the shapefile for each street, a street containing more damage was found and a subset of this street and associated images were selected as the ZEB image data set.

The second step is road damage detection. The Global Road Damage Detection Challenge 2020 data set was used as the training set to train the model. The training model is then used to detect damage in the ZEB data set.

The third step is the scoring model. The bounding box data is obtained from the detection model to create a scoring data set. Finally, a regression model is selected for scoring

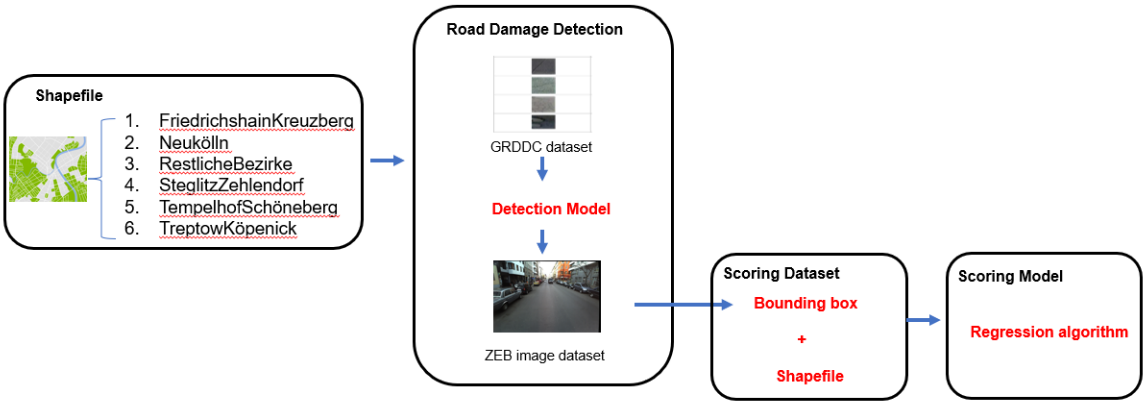


Figure 2: Research process map

2 Shapefile

This chapter describes the creation of ZEB data set from shapefile.

2.1 Theory of Shapefile

The shapefile format is a geospatial vector data format for geographic information system (GIS) software for storing geographic location and associated attribute information. This format lacks the capacity to store topological information. The shapefile format was introduced with ArcView GIS version 2 in the early 1990s. It is now possible to read and write geographical datasets using the shapefile format with a wide variety of software. [2]

The shapefile format stores the data as primitive geometric shapes like points, lines, and polygons. These shapes, together with data attributes that are linked to each shape, create the representation of the geographic data. The term "shapefile" is quite common, but the format consists of a collection of files with a common filename prefix, stored in the same directory. The three mandatory files have filename extensions .shp, .shx, and .dbf. The actual shapefile relates specifically to the .shp file, but alone is incomplete for distribution as the other supporting files are required. [1]

Figure 3: Shapefile of Berlin

2.2 Creation of The ZEB Dataset

Shapefiles for six streets in Berlin are provided. They are FriedrichshainKreuzberg, Neukoeln, RestlicheBezirke,SteglitzZelendorf, TempelhofSchoeneberg and TreptowKoepeniok. Each street is divided into primary and secondary roads. As shown in the Fig.4, based on the attribute and values of the shapefile, it can be seen that the primary roads of FriedrichshainKreuzberg has more damage and using the primary roads of the FriedrichshainKreuzberg for damage detection works better. Therefore, subset of the FriedrichshainKreuzberg primary roads was selected as the ZEB image data set. After the ZEB image data set has been selected, the image data set first needs to be pre-processed. Firstly all images need to be reset to a uniform size of 640x640 and secondly all images need to be renamed in order and placed in the same folder in preparation for damage detection.

fid_1	bezeichnu	gis_id	elem_nr	datum	bildpfad	pgr_m	pgr_a	zw_pgr_m	zw_pgr_a	au
2358	Fahrbahn	Fl_201504	45510044_	######	Los1\Bildc	14.75	3.495461	4.19	4.5	
2356	Fahrbahn	Fl_201504	45510044_	######	Los1\Bildc	14.27	3.947835	4.07	4.95	
2364	Fahrbahn	Fl_201504	45510040_	######	Los1\Bildc	5.74	2.25396	1.94	3.17	
2365	Fahrbahn	Fl_201504	45510040_	######	Los1\Bildc	10.76	3.669091	3.19	4.67	
2328	Fahrbahn	Fl_201504	45510025_	######	Los1\Bildc	10.99	4.418667	3.25	5	
2329	Fahrbahn	Fl_201504	45510025_	######	Los1\Bildc	8.98	3.014043	2.75	4.01	
2331	Fahrbahn	Fl_201504	45510025_	######	Los1\Bildc	5.96	1.82	1.99	2.59	
2339	Fahrbahn	Fl_201504	45510025_	######	Los1\Bildc	19.27	5.438492	5	5	
2333	Fahrbahn	Fl_201504	45510025_	######	Los1\Bildc	9.61	2.522706	2.9	3.52	
2334	Fahrbahn	Fl_201504	45510025_	######	Los1\Bildc	15.95	2.942258	4.49	3.94	
2393	Fahrbahn	Fl_201504	46520019_	######	Los1\Bildc	16.49	4.045152	4.62	5	
2394	Fahrbahn	Fl_201504	46520019_	######	Los1\Bildc	6.11	2.512	2.03	3.51	
2396	Fahrbahn	Fl_201504	46520019_	######	Los1\Bildc	19.31	1.907833	5	2.71	
2392	Fahrbahn	Fl_201504	46520019_	######	Los1\Bildc	12.94	3.374833	3.74	4.37	

Figure 4: Attribute and values of the shapefile

Figure 5: Road of FriedrichshainKreuzberg

3 Road Damage Detection

3.1 Model Selection

As we all known,there are currently plenty of model involved in image detection.In this project,we just analyze and do some comparisons among the three most popular models,namely SSD,Faster-RCNN and YOLOv5.

3.1.1 SSD

SSD (Single Shot Multibox Detector) is a single-level and multi-layer detection model.Because it only performs frame prediction and loss calculation once, it belongs to one-stage algorithm.
SSD adopts a single CNN structure to perform multiple bounding box predictions in a one-stage mode. Specifically, SSD predicts the position transformation parameters and target category scores based on the default bounding boxes in the form of small window convolution for the main CNN structure at different levels of feature maps, so as to obtain bounding box in different scales, different shapes.
In fact,SSD tends to hand over the task of small target detection to shallow features, but shallow features often show more texture information, and the expression of semantic information is not enough, so the detection accuracy for small targets is not very good.

3.1.2 Faster-RCNN

Based on R-CNN and Fast RCNN, Ross B. Girshick proposed Faster R-CNN(Fast Region-based Convolution Neural Networks) in 2016. Using VGG16 as the backbone of the network, the speed reached 5fps on the GPU, and the accuracy rate was further improved. Faster RCNN abandons the traditional sliding window and selective search methods, and directly uses RPN to generate detection frames. This is also a huge advantage of Faster RCNN, which can greatly improve the generation speed of detection frames.
The RPN (Region Propose Network) module processes the feature maps to find a predefined number of regions (bounding boxes) that may contain objects. The main process performed by the RPN module is to input the public feature map of Faster-RCNN, generate anchors, extract and fine-tune the positive sample anchors through the activation function SoftMax, and the final proposal layer is responsible for synthesizing the positive anchors and the corresponding bounding box regression offset to obtain proposals, and at the same time eliminate overlapping and out-of-bounds proposals.
Faster-RCNN improved detection accuracy and speed compared with RCNN and Fast-RCNN.It only takes 10ms to generate a proposal region.But it still needs to obtain the region proposals,and then classify each Proposal.Therefore it is computationally intensive and time-consuming.

3.1.3 YOLOv5

YOLO treats the object detection task as a regression problem.With YOLO, each image only needs "one look" to find out which objects are in the image and the positions of these objects.it is a one-stage algorithm, using one CNN directly to detect and classify the object,which is faster.At the same time, Image features extracted by YOLO are more general. Beside, Yolo is more suitable for embedded development,like if we wanna apply it on the phone. It needs less capacity than other model.
In consideration of our computer configuration and the general performance of these three model,we finally select yolov5 as our detecction model.

3.2 Datasets

The image dataset provided by the IEEE Big Data 2020 Global Road Damage Detection Challenge was collected from three countries: Czech Republic (CZ), India (IN), and Japan(JP). The training datasets is composed of 21,041 images,and each image is annotated by one or more classes of road damages; Some of the training images are not annotated with any road damage class; thus they are free from any road damages. In this paper,we considered only the images annotated by four specific classes of road damages (namely,longitudinal Crack,Transverse Crack,Alligator Crack and Pothole) shown in Figure 7 and the remaining images were considered as free of road damages when designing our solution.
The statistics for the distribution of images in the three datasets and the three countries are provided in Figure 6. The training data contains annotation with marked labels for different road damage types.

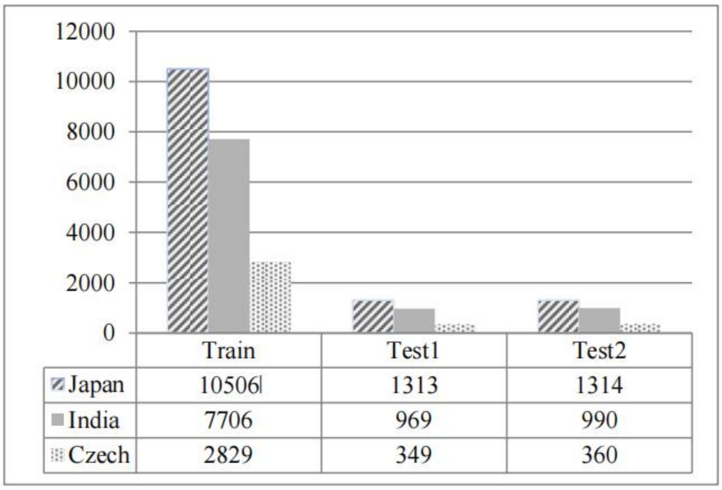


Figure 6: Dataset Statistics

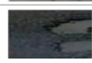
Pavement Distress	Distress Class	Sample Image
longitudinal Crack	D00	
Transverse Crack	D10	
Alligator Crack	D20	
Pothole	D40	

Figure 7: Road Damage Classes

3.3 Preprocessing

First of all,we delete the images and annotations which do not possess any labels.Then because the format of the datasets GRDDC provided is in PASCAL VOC format.So After we split the datasets into train and validation sets with split ratio=0.8, we convert the PASCAL VOC format to YOLO format.

Japan_000003.txt

1	2 0.61 0.8558333333333333 0.7166666666666667 0.2883333333333333
2	2 0.04583333333333334 0.895 0.085 0.14666666666666667

Figure 8: YOLO Format

3.4 Training

The training process is based on Linux 20.04 with GPU NVIDIA GTX 2080Ti.

3.4.1 Configuration

Figure 9 shows the configuration of road detection model.

1	Cython
2	matplotlib>=3.2.2
3	numpy>=1.18.5
4	opencv-python>=4.1.2
5	pillow
6	PyYAML>=5.3
7	scipy>=1.4.1
8	tensorboard>=2.2
9	torch>=1.6.0
10	torchvision>=0.7.0
11	tqdm>=4.41.0

Figure 9: Configuration Requirement

3.4.2 Training Parameters

We trained the models based on yolov5s and yolov5x and their corresponding weights that provided by Ultralytics open-source [3]. Figure 10 shows some parameters of the training model including Hyper-parameters.

1	weights	yolov5s.pt/yolov5x.pt
2	cfg	yolov5s_road.yaml/yolov5x_road.yaml
3	data	road2020.yaml
4	epoch	300
5	batch-size	8
6	img-size	[640, 640]
7	device	cuda0
8	optimizer	adam
9	workers	8
10	resume	FALSE
11	number of classes	4
12	initial learning rate	1.00E-03

Figure 10: Parameters

3.5 Model Comparison and Conclusion

After training, we evaluate the models on validation set and did some comparison between Yolov5s and Yolov5x.As we can see in the confusion matrix of Yolov5s(Figure 11 and Figure 12),The data on the diagonal represents the rates of correct prediction.Apparently,The predictio accuracy of Yolov5x is far better than Yolov5s,which are all around 80%.But the detection on damage type D10(namely Transverse Crack) perform relatively bad(only 59%),one of the reason might be the lack of D10 damage in Training datasets.
Beside,referring to their corresponding PR curve shown in Figure 13 and Figure 14,the mAP of Yolov5x in all classes reach ***0.76*** which is relatively acceptable. However,it takes ***66.153s*** to finish the validation process while using Yolov5s model but it take ***426.556s*** by Yolov5x.
Therefore, we can draw a conclusion that Yolov5x has high accuracy,but it is more time-consuming than v5s. On the other hand, YOLOv5s has faster detection speed, which is over 6 times faster than Yolov5x here. So it require less computation and can better meet real-time requirements,like detection on video. and because of small capacity, it is easier to deploy on mobile terminals.In the mean time,it shows that this Yolov5 model perform much better on Alligator Crack(D20) detection but not suitable for Transverse Crack(D10).In this project, we used yolov5x as our final model because we need to guarantee the detection accuracy instead of speed.

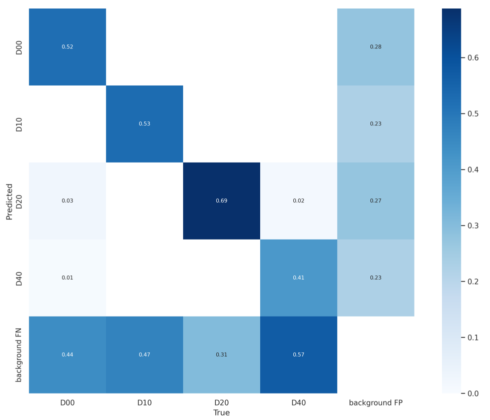


Figure 11: Confusion Matrix of Yolov5s

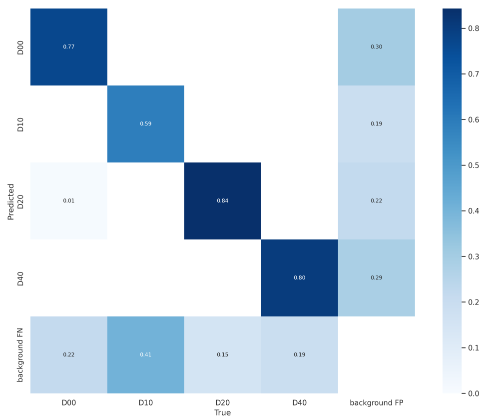


Figure 12: Confusion Matrix of Yolov5x

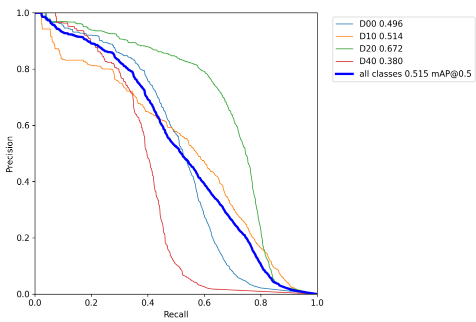


Figure 13: PR curve of Yolov5s

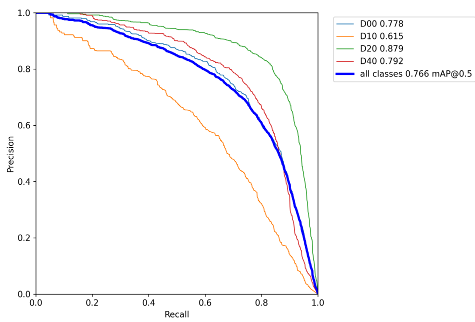


Figure 14: PR curve of Yolov5x

3.6 Detection Results Analysis

In order to find out how this model perform on ZEB-datasets and continue our road damage rating model,we used the trained Yolov5x model to detect the road damage based on ZEB-datasets.

3.6.1 Detection Hyper-parameter

Despite the default parameter,we adjust confidence-threshold and iou-threshold to fit our rating model.In consideration of avoiding missing any road damage for the reason that tiny road damages are also involved in rating process after we analyze the shapefile, we set the confidence-threshold to ***0.10***. Meanwhile,we finally decide the iou-threshold to be ***0.30*** in order to avoid duplicate detection frame.

3.6.2 Documentation Output Transformation

After detection,we got the documentation in Yolo format including the position as well as the confidence of the road damage. But in the damage rating regression model later on, the area of the damage as well as the diagonal line of the bounding box are set as one of the input,so we decided to convert the documentation into a format that could easier handle. After analyze the ZEB image,we realize the image size is always 1002x668,so we convert the the output data into format (positions of upper-left and bottom-right within height:1002 and width:668).

3.6.3 Detection Results Visualization

Figure 15 and Figure 16 illustrate the detection result of ZEB-data which is acceptable. Almost all the road damage could be detected.

Figure 15: Original Image

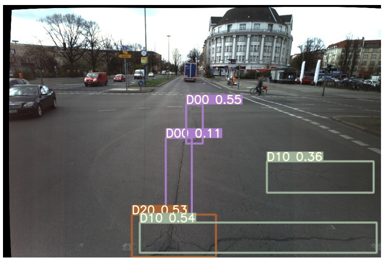


Figure 16: Predicted image

3.6.4 Detection Error Analysis

However, there are still different kind of factors that influence the detection,leading to the Detection error.For example,the shadow of the tree which shown in Figure 17,would be recognized as Alligator Crack(D20) with high confidence. And the gravel roads which is considered as the special feature of Berlin, are also mispredicted as Alligator Crack(D20). These kind of detection errors could not be avoided during the detection process for now. But we did some data cleansing process in order to better fit the road damage scoring model,namely denosing.In fact,we find out Alligator Crack is usually small,so we iterate over all results detected as D20,and remove the results when the area of the bounding box is too large.There are also detection error caused by bicycle lane sign and low image resolution which are illustrated in Figure 19 and Figure 20.In this paper we consider these as the inevitable problems.

Figure 17: Error caused by Shadow

Figure 18: Error caused by Gravel Roads

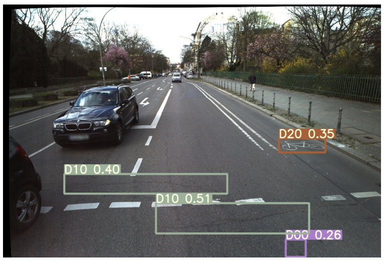


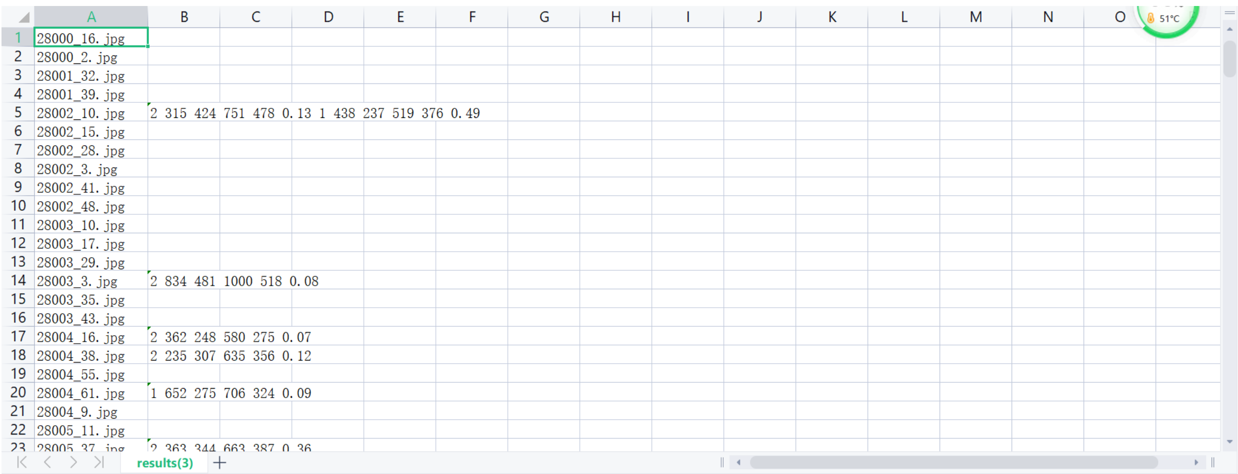
Figure 19: Error caused by bicycle lane sign

Figure 20: Error caused by low image resolution

4 Scoring Model

4.1 Data Preprocessing

After getting the detected data from detection model, the detected score looks like, the first

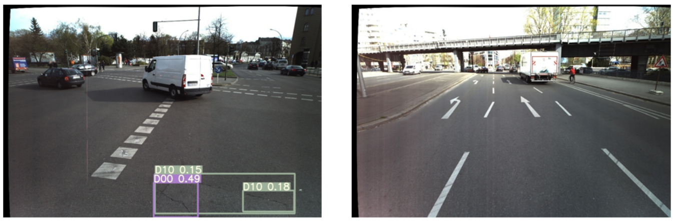


	A	B
1	28000_16. jpg	
2	28000_2. jpg	
3	28001_32. jpg	
4	28001_39. jpg	
5	28002_10. jpg	2 315 424 751 478 0. 13 1 438 237 519 376 0. 49
6	28002_15. jpg	
7	28002_28. jpg	
8	28002_3. jpg	
9	28002_41. jpg	
10	28002_48. jpg	
11	28003_10. jpg	
12	28003_17. jpg	
13	28003_29. jpg	
14	28003_3. jpg	2 834 481 1000 518 0. 08
15	28003_35. jpg	
16	28003_43. jpg	
17	28004_16. jpg	2 362 248 580 275 0. 07
18	28004_38. jpg	2 235 307 635 356 0. 12
19	28004_55. jpg	
20	28004_61. jpg	1 652 275 706 324 0. 09
21	28004_9. jpg	
22	28005_11. jpg	
23	28005_37. jpg	2 363 344 663 387 0. 36

results(3)

Figure 21: Data preprocessing

column describes the number of images, and the following are the types of damage , the left upper corner of the damage, right down corner of the damage and the confidence level of the damage.For instance, (2 315 424 751 478 0.13) two indicates the damage type, (315 424 751 478) means x1,y1,x2,y2 of the coordiantion of the damage, 0.13 is the confidence level of the damage. In the next stage, we implemented some simple data preprocessing job, like dataset cleaning by denoising, for example deleting significant damage but twrio and zwris values are 1, as well as twrio and zwris values are large but damage is insignificant.



twrio=1; zw ris = 1 twrio = 2.93; zw ris = 2.79

Figure 22: Data Cleaning

4.2 Dataform Transformation

Attributes of the shapefile are collected in one csv file.which contains main information bildpfad,twrio and zwris. As we defined in the crack type part: there are four crack types categories: longitudinal crack, transverse crack, alligator crack and pothole. For the first two categories there exist sometimes the cases the crack are exactly the diagonal of the size. At the same time for the next two categories the area represents more precisely of the real damage degree.Therefore the dataset is transformed into to two versions: diagonal version and area version. Method one to calculate the diagonal and method two to calculate the area:

$$\text{diagonal} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Figure 23: diagonal

$$area = (x_2 - x_1)(y_2 - y_1)$$

Figure 24: area

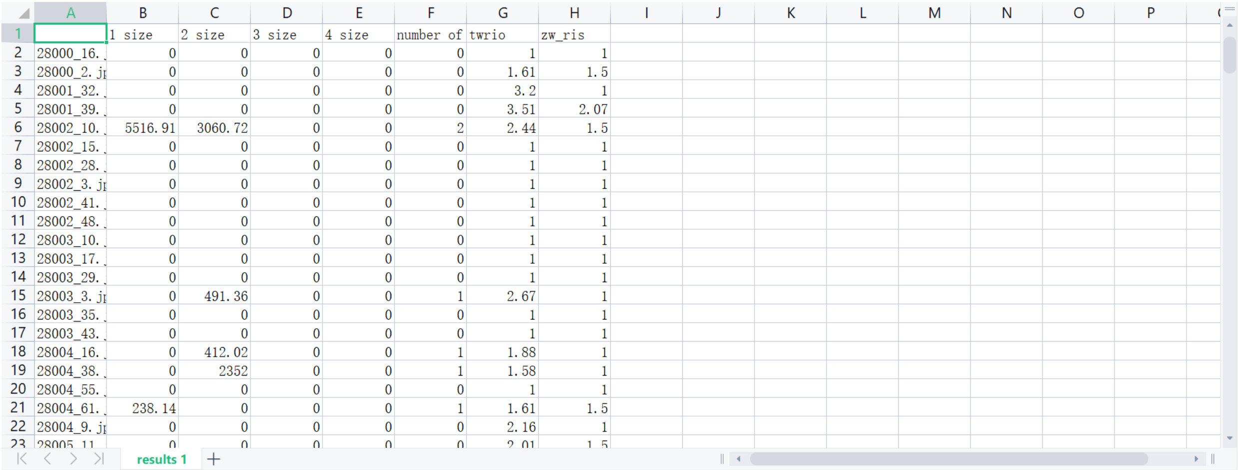
First column indicates the number of the picture, the following are the size of 4 type categories damage, number of damages in the picture, and two target values twrio and zwris. Diagonal version:

In the diagonal version, first column indicates the number of the picture as well, the following are diagonal of 4 type damages, and two target values twrio and zwris.

4.3 Scoring Model

The scoring model is applied to figure out the relationship between shapefile attribute and detecting result.Regarding the scoring model we implemented two regression models.

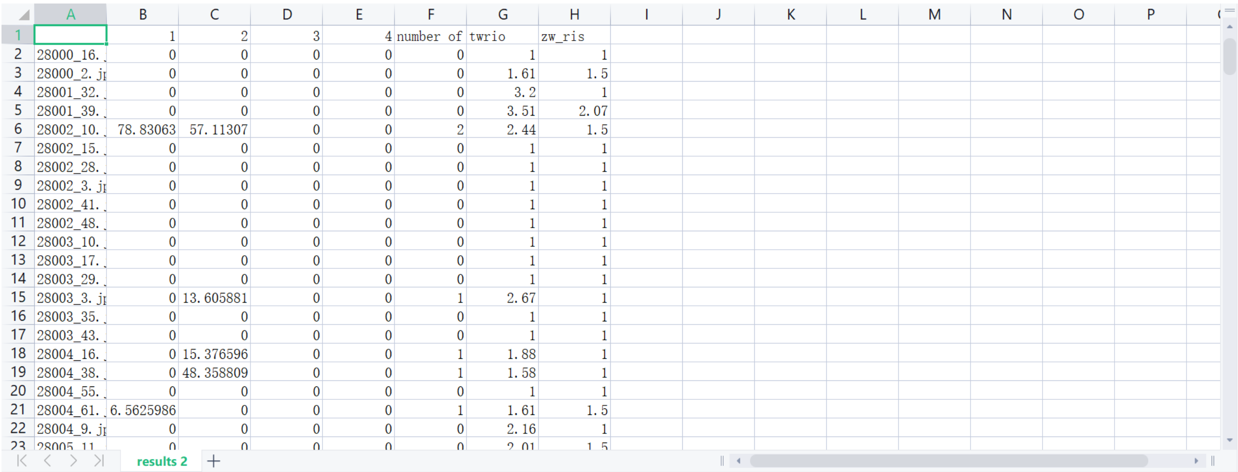
1. Random forest Random forest is a supervised learning algorithm , which is an integrated learning algorithm based on decision tree algorithms. "Random" means that the data set and features are randomly selected. In the Random Forest process, we take bagging with replacement from the original data set to construct a sub-data set. The data volume of



	A	B	C	D	E	F	G	H
1		1 size	2 size	3 size	4 size	number of	twrio	zw_ris
2	28000_16.	0	0	0	0	0	1	1
3	28000_2. j	0	0	0	0	0	1.61	1.5
4	28001_32.	0	0	0	0	0	3.2	1
5	28001_39.	0	0	0	0	0	3.51	2.07
6	28002_10.	5516.91	3060.72	0	0	2	2.44	1.5
7	28002_15.	0	0	0	0	0	1	1
8	28002_28.	0	0	0	0	0	1	1
9	28002_3. j	0	0	0	0	0	1	1
10	28002_41.	0	0	0	0	0	1	1
11	28002_48.	0	0	0	0	0	1	1
12	28003_10.	0	0	0	0	0	1	1
13	28003_17.	0	0	0	0	0	1	1
14	28003_29.	0	0	0	0	0	1	1
15	28003_3. j	0	491.36	0	0	1	2.67	1
16	28003_35.	0	0	0	0	0	1	1
17	28003_43.	0	0	0	0	0	1	1
18	28004_16.	0	412.02	0	0	1	1.88	1
19	28004_38.	0	2352	0	0	1	1.58	1
20	28004_55.	0	0	0	0	0	1	1
21	28004_61.	238.14	0	0	0	1	1.61	1.5
22	28004_9. j	0	0	0	0	0	2.16	1
23	28005_11	0	0	0	0	0	2.01	1.5

results 1

Figure 25: Area version dataset



	A	B	C	D	E	F	G	H
1		1	2	3	4	number of	twrio	zw_ris
2	28000_16.	0	0	0	0	0	1	1
3	28000_2. j	0	0	0	0	0	1.61	1.5
4	28001_32.	0	0	0	0	0	3.2	1
5	28001_39.	0	0	0	0	0	3.51	2.07
6	28002_10.	78.83063	57.11307	0	0	2	2.44	1.5
7	28002_15.	0	0	0	0	0	1	1
8	28002_28.	0	0	0	0	0	1	1
9	28002_3. j	0	0	0	0	0	1	1
10	28002_41.	0	0	0	0	0	1	1
11	28002_48.	0	0	0	0	0	1	1
12	28003_10.	0	0	0	0	0	1	1
13	28003_17.	0	0	0	0	0	1	1
14	28003_29.	0	0	0	0	0	1	1
15	28003_3. j	0	13.605881	0	0	1	2.67	1
16	28003_35.	0	0	0	0	0	1	1
17	28003_43.	0	0	0	0	0	1	1
18	28004_16.	0	15.376596	0	0	1	1.88	1
19	28004_38.	0	48.358809	0	0	1	1.58	1
20	28004_55.	0	0	0	0	0	1	1
21	28004_61.	6.5625986	0	0	0	1	1.61	1.5
22	28004_9. j	0	0	0	0	0	2.16	1
23	28005_11	0	0	0	0	0	2.01	1.5

results 2

Figure 26: diagonal version dataset

the sub-data set is the same as that of the original data set. Elements in different sub-datasets can be repeated, and elements in the same sub-data set can also be repeated. In addition, each splitting process of the sub-tree in the random forest does not use all the candidate features, but randomly selects certain features from all the candidate features, and then selects the optimal feature from the randomly selected features. Random forest is proven to have the advantages of easy implementation, low computational cost, and excellent performance in classification and regression problems.

2. support vector regression "Support Vector Machine" (SVM) is a supervised machine learning algorithm that can be used for both classification or regression challenges. In the SVM algorithm, each data is drawn as a point in an n-dimensional space (where n is the number of features), and the value of each feature is the value of a specific coordinate. Then, classify the data by finding a hyper-plane that can distinguish the two classes well. Sup-

port Vectors are simply the coordinates of individual observation. The SVM classifier is a frontier that best segregates the two classes (hyper-plane/ line). The same, support vector regression is to find the hyperplane in oder to minimize the distance of the data to the hyperplane.

3. Particle Swarm Optimization The core idea of the PSO algorithm is to find the global optimal solution through information sharing between particles. It is a random search algorithm based on group cooperation, developed by simulating the foraging behavior of birds.The bird flocks share information in the search process so that other birds know the location of the largest food, and finally, all the birds gather around the largest food to reach the optimal solution. The outcome of the algorithm is to optimized particles in the D dimension space, and each particle is searched along the direction of the current optimal solution, and the information is shared between the particles, and finally, the optimal solution is found when the global convergence is reached.

4. Parameters of PSO There are some pre-set hyperparameters in PSO, when writing code: • w: Initial weight of the particle velocity, reduce at each iteration. • c1; c2: Velocity parameters for global optimal and particle optimal at each iteration. • [Vmin; Vmax]: Maximum and minimum velocity of particles. • Dim: The number of parameters that the objective function contains that needs to be optimized. • maxgen: The number of the iterations. • sizepop: The number of the particles, will affect the speed of convergence and computation time. • fitness: The objective function which is supposed to be optimized.

5. Function of PSO

Particles update their speed and position through the following formulas: pbest represents

$$\mathrm{v}_{\mathrm{id}}^{\mathrm{k}} = w \cdot v_{id}^{k-1} + c1 * \mathrm{r}1(\mathrm{pbest}_{\mathrm{id}} - x_{id}^{k-1}) + c2 * \mathrm{r}2 \cdot (gbest_{id} - x_{id}^{k-1})$$

Figure 27: picture1

$$x_{id}^{k} = x_{id}^{k-1} + v_{id}^{k}$$

Figure 28: picture2

the personal best of the particle(the best cost in all iterations) and gbest represents the global optimal for all particles of all iterations. r1 and r2 are tow random numbers. indicate the position of the particle, which is calculated from the previous step's position and the velocity from last equation. According to the velocity update formula, the velocity of a particle consists of three parts: The first part is the inheritance of the particle's previous speed, which reflects the inertia of particle motion; the second part is self-cognition, which indicates the influence of the particle's own previous flight experience on the subsequent flight direction; the third part is social cognition, which represents the influence of the flight experience of all particles in the population on the subsequent flight direction of each particle.

4.4 Result

With two regression models: Random Forest Regressor and Support Vector Regressor; two Dataset: area dataset and diagonal dataset; two target values: twrio zwris. We achieved eight

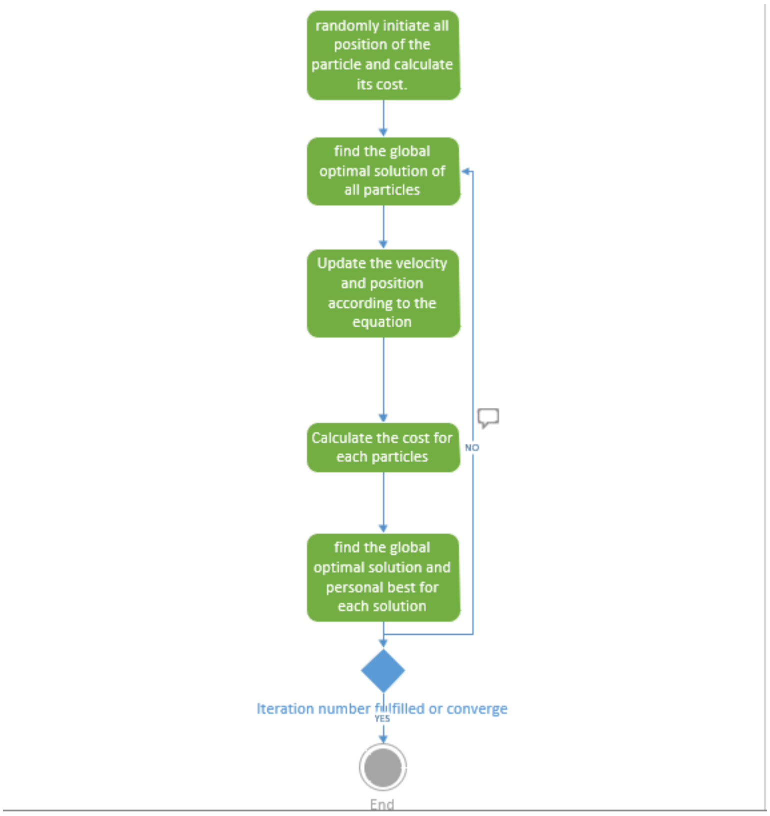


Figure 29: workflow of pso

pairs of result in the end, from which Random Forest Regressor with area dataset and twrio achieved the best score(MSE 0.4196 and R-Squared 0.5657). In the eight cases, twrio win in the score compared to zwris.

Model	Dataset	MSE	R-Squared
Random Forest Regressor	Dataset 1 with TWRIO	0.4196	0.5657
	Dataset 1 with ZW_RIS	0.2535	0.4435
	Dataset 2 with TWRIO	0.4926	0.5503
	Dataset 2 with ZW_RIS	0.21949	0.5180
Support Vector Regressor	Dataset 1 with TWRIO	0.4839	0.4935
	Dataset 1 with ZW_RIS	0.2700	0.4000
	Dataset 2 with TWRIO	0.6651	0.30387
	Dataset 2 with ZW_RIS	0.2824	0.3800

Figure 30: result of regression model

5 Summary and Outlook

To figure out which dataset represents more of the truth of the damage, we calculated two version dataset, one to calculate the area of the damage, and one to calculate the diagonal of the damage, the result shows the area has the lower MSE and higher R-Squared. Among the two regressor we implemented, random forest regressor achieve the best score in the all cases, Twrio achieve the higher R-Squared in most cases. For further improvement, a larger dataset with more real detected value as well as pre-processing technique is in demand.

References

[1] Shapefile. https://www.e-education.psu.edu/natureofgeoinfo/book/export/html/1726, Last accessed on 2022-07-25.

[2] Shapefile, ArcGIS Online help. https://doc.arcgis.com/zh-cn/arcgis-online/reference/shapefiles.htm, Last accessed on 2022-07-25.

[3] Glenn Jocher, Ayush Chaurasia, Alex Stoken, Jirka Borovec, NanoCode012, Yonghye Kwon, TaoXie, Jiacong Fang, imyhxy, Kalen Michael, Lorna, Abhiram V, Diego Montes, Jebastin Nadar, Laughing, tkianai, yxNONG, Piotr Skalski, Zhiqiang Wang, Adam Hogan, Cristi Fati, Lorenzo Mammana, AlexWang1900, Deep Patel, Ding Yiwei, Felix You, Jan Hajek, Laurentiu Diaconu, and Mai Thanh Minh. ultralytics/yolov5: v6.1 - TensorRT, TensorFlow Edge TPU and OpenVINO Export and Inference, February 2022.

List of Figures