Robust Object Detection Model for Autonomous Driving in Real Scenarios

Ningchan Wang

Jurong Country Garden School, Jiangsu, 210007, China

Abstract: *This article explains the method of optimizing the object detection model used in the automatic drive. At present, there are a lot of car accidents happen caused by the low robustness of the automatic pilot. We are focused on improving the robustness of the model in a unique environment. We built four special environments and collected the images of the environment in a special way, and used them to train the detectron2, which is our initial model. Adversarial training is the way to train the model. Finally, we optimize the model successfully. If more people are willing to spend time and money training a particular model for a location, the automatic pilot can be promoted to any place.*

Keywords: *Object detection, Adversarial training, Feature Pyramid Networks*

1. Introduction

The automatic drive is an emerging application of artificial intelligence that has been rapidly developed in recent years. In 2011 google first showed the prototype of the automatic pilot. And in the next year, the automatic pilot has tested on a highway in California. Two years later, the Beijing Municipal Commission of Transportation and Beijing Municipal Public Security Traffic Management Bureau and Beijing Municipal Economic Information Commission, and other departments published legislation about automatic driving which is an indication of introduce of the automatic pilot in China.

The society of automotive engineers (SAE) formulates a set of systems that describe automatic drive in different stages, benchmarking different functions of automatic drive. In this system, automatic drive are standardized into 5 levels. Level 5 required automatic pilot fully autonomous driving. No driver is needed to monitor the road condition or the driven route ·

Nowadays, the automatic pilot still stays at level 4, which means people only need to monitor the decision that artificial intelligence (AI) makes. The reasons for the standstill in the development of automatic drive are. First, the function of the automatic pilot has low robustness. In march 18th, 2018, a woman was hit by a car in Arizona (USA), there are a lot of cases caused by automatic pilot, which causes the percentage of supporting automatic pilot on average, 29% (Auto Pacific interview about FSD). Second, most people reject automatic pilots, because of the danger of Privacy disclosure. Third, from a humanitarian point of view, the unemployment of society causes the mark time of the automatic drive.

The core technology of automatic drive are decision-making, navigation, communications security and object detection. Most of these technologies are mature and perform an excellent function in the laboratory. But once the automatic pilot faces an environment that has a slight difference, the automatic pilot will expose poor transferability and generalizability. This quality restricts the large-scale deployment of automatic pilots. June 15th, 2022, NHTSA published two reports about the road accident caused by automatic drive. In the report, there are 392 accidents occur when the driver is assisted by the L2 Advanced driver assistance system and 130 accidents occur when vehicles carry the L3-L5 Advanced driver assistance system. The report also shows under the premise that 75% of accidents didn't publish casualties, and 12 people were serious injuries. Most accidents are caused by low robustness. The automatic pilot has a long way to go.

Although there are lots of advancing models studied in laboratories [1], there is still a huge gap between lab results and practical uses. Automatic pilots are facing these challenges, first challenge is the accuracy of 3D semantic segmentation decreases, the second challenge is not enough robustness, the third challenge is fewer heterogeneous multi-sensor fusion, last challenge is not all weather conditions are included. Most of the challenges are caused by less quantity of training samples, the high cost of data collection, unbalanced training data, and incomplete scene coverage. To fulfill the detection model for practical uses and overcome challenges, in this paper, we construct a real experimental environment and collect data from the trimming model to improve the robustness of the model, to prevent accidents from happening.

2. Related work

2.1. Object detection

Object detection is the key technology of advanced driver assistance systems (ADAS). It ensures the safety of the passenger. It automatically recognizes objects of a particular class by using two important technology computer vision and image processing. Face detection and pedestrian detection are the most popular in object detection[2]. Generally, there are two main approaches to achieving autonomous driving, Lidar and computer vision. Lidar is often used in the automatic drive. This technology uses the time of the reflecting laser to determine the distance between the laser emitter and the surface of the object. Lidar play an important role in object detection, it provides a bigger range of three-dimensional measurement for the object. Although Luminar's Lexus has proof of the importance of Lidar, Musk believes that a vision-based system is the only way, he said that "LIDAR is a fool's errand." Building object detectors which have a utilizing computer vision, the first thing is designing a network architecture that is able to learn the features of the class of a specific object, then collect a large number of labeled data. Finally, train the model with data. The system is shown in figure 1.

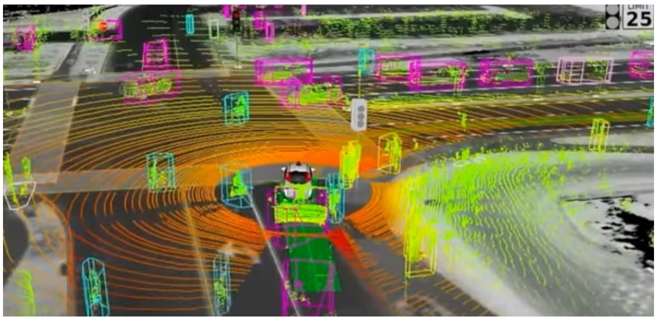
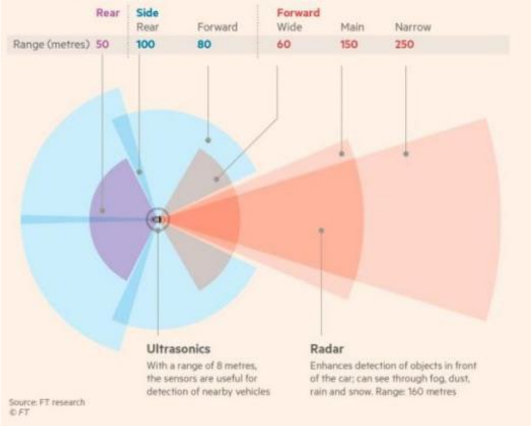


Figure 1: Effect picture of vision-based approach (without Lidar and HD maps)

Tesla vehicles use eight cameras, or sensors, to identify and recognize real-world objects, which is shown in figure 2. These cameras collect data of animals, pedestrians, vehicles or others, to prevent accident happen on drivers or others. This function requires high accuracy of object detection and real-time detection with low latency. Tesla can perfectly handle this job without LIDAR and HD maps because LIDAR and HD maps are not scalable and too slow. This is why we use a vision-only system.



With a range of 8 metres, the sensors are useful for detection of nearby vehicles

Enhances detection of objects in front of the car; can see through fog, dust, rain and snow. Range: 160 metres

Source: FT research
© FT

Figure 2: How Tesla's eight camera works.

Tesla's object detection can't achieve success without a powerful database. The team of Tesla manufactures a special AI for collecting data. This AI can train models with this task. First, label an original dataset with new object classes. Then train the model with new data and test the new model, after that, AI judges the result of the training. Lastly, AI searches for the cases which don't output satisfying and fetches for a similar case and retrains. This efficient AI collected data for more than 3 billion miles for an automatic drive.

2.1.1. R-CNN

Region Based Convolutional Neural Networks (R-CNN) is an object detector. R-CNN was developed to produce a set of bounding boxes on the input image which contain the information of the object that is bounding, such as the class of the object and it is a machine learning model, especially for computer vision.

In the year of 2013, R-CNN applied a mechanism called selective search. In the year 2015, R-cnn upgraded to fast R-cnn, which runs the neural network once on the whole image. In the same year, Faster R-cnn has released, and this version of R-CNN integrates the ROI generation into the neural network. Mask R-cnn was published in 2017, it adds instance segmentation. In 2019 the researcher released mesh R-cnn, which has the ability to generate a 3D mesh from a 2D image.

2.1.2. Detectron2

Detectron's idea was first started in 2008. When meta-AI research Scientists wanted to make their object detection research possible. The start of Object detection research means that it can be transplanted into a computer version. In the year of 2012, R-CNN have developed. R-CNN is an object detector with deep learning. The researcher of detectron witnessed a different version of CNN arises: Faster R-CNN, Mask R-CNN and others. The witness make them realize the importance of implementing, benchmarking and reproducing ideas for the community.

Detectron was release in 2018. It is widespread on the internet and has become on of Facebook AI Research (FAIR)'s most adopted open-source projects. Detectron2 is the model, which is rewritten version of detectron and started with mask rcnn-benchmark. The platform is now implemented in PyTorch. Detectron2 provide fast training on servers with a single GPU or server with multiple GPUs. Which provide more flexibility algorithms: DensePose, panoptic feature pyramid networks, mask R-CNN. Detectron2 shows a similar function to mask R-CNN.

2.2. Threats and Adversarial Examples

2.2.1. Adversarial examples generating

The adversarial example has a slight difference from the original data, which causes the machine-learning model to make a wrong prediction, which deceives the model. Figure 3 illustrates the cases. The reason for using adversarial examples is that we attack the model to expose its weakness of the model in order to make it less vulnerable[3].

The adversarial example has intentional feature perturbations, these noise can be made in a different method. The first method is fast gradient sign method (FGSM).

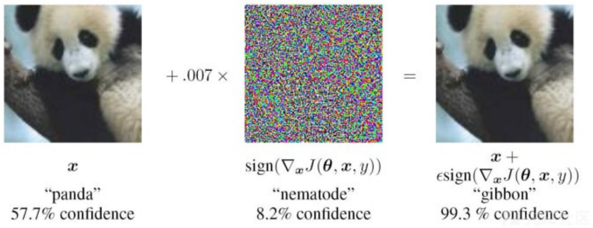


Figure 3: Wrong result of detection cause by noise (Adversarial examples) .

X is the original sample,θ is the important parameter, and y is the true class of x. Input original sample importance parameter and true class using the loss value of the neural network (J) to find the adversarial example .

$$\eta =\in sign(\nabla_x J(\theta, x, y))$$

A stop sign may be mistaken as a slow-down sign if the noise is added to the original image, which causes a big safety trouble in the automatic drive. Although the noise can't be recognizance by human vision, it can cause serious misinterpretation. So adversarial example used in the automatic drive is necessary.

2.2.2. Adversarial training

The idea of Adversarial training was first brought up by lan j. Goodfelliw and the team, in the year 2014, many machine learning models. In 2015 Adversarial training offered an easy way to train.

Adversarial training is a very important way to improve robustness. In the progress of adversarial training, the sample will be append similar sample which has a tiny difference but maybe cause an incorrect result. Using this sample to train the model. So, the model improves robustness.

2.2.3. Transfer learning

Deep-learning object detectors often require a database. Many large open-source datasets, such as imageNet, which has 14 million images in 22000 classes. Using this database to train deep learning model costs a lot. So we need a more efficient way to train deep learning models. Transfer learning makes deep learning training much less demanding.

Neural network development use composed of multiple layers. For example, detect characteristics of an object: edges, colors or others, then detect the class of the object. When the model uses transfer learning, engineers freeze the first layers of the neural network, then trim the deeper layers with new layers and new classes about the database[4].

3. Automatic drive system

3.1. Dataset Preparation

Object detector often has low robustness. The database of Tesla may be only suitable for California, it exposes its low robustness when having accidents in another place. So we are preparing a task-specific dataset in the real-world application environment.

In particular, we built 4 experimental simulated reality environments, which is shown in table 1.

Table 1: Characteristic of different scene.

	Office Of teacher	Hall Of cafe	Road In school	Bicycle Parking lot
Main class	People Chair Table Box plant	People Chair laptop	Vehicle people	Bicycle people
light	Indoor light	Half-indoor Half-outdoor	Sunny outdoor	Cloudy outdoor

These environments include two scene types, outdoor and indoor. These four environments have a different characteristics, which has absolutely representative. The first scene is an office, which is shown in figure 4, which is an indoor scene and, it comprises 5 different classes of objects person, a chair, a table, a box and plants, also the camera of the robot absorbing one way indoor light from windows.

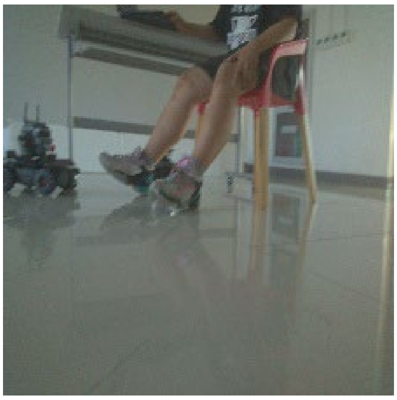


Figure 4: Example of picture of office

The second scene is the hall, which is shown in figure 5. The hall includes three classes which are person, chair and laptop, the category of light is different from the office because the hall is half-transparent, so the camera collects the image in the condition of half indoor light half out door light.

Figure 5: Example of picture of Hall of cafe

The third scene is the street for vehicles, which is shown in figure 6. The most important part of training is to test the sensitivity of the car of the model, which is included is the scene, when taking a photo of a real street pedestrian can't avoid. The camera takes fully sunny outdoor light in this scene.

Figure 6: Example of picture of Road in school

The fourth scene is the parking place for bicycles, which is shown in figure 7. This scene includes the next important class of this training: bicycles and pedestrians can't be eliminated. In this scene, the camera takes pictures on a cloudy day and in fully outdoor light.

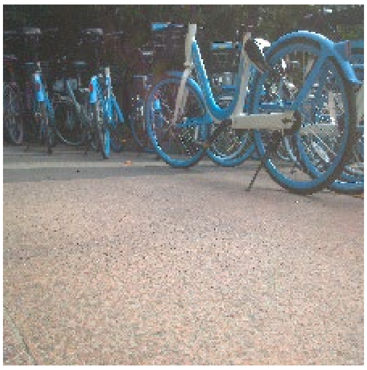


Figure 7: Example of picture of Bicycle parking lot

The orbit of robot is special for data collection. Which is shown in figure 8. When thinking about collecting data from the environment, the first orbit that came up on my mind is rotation which can collect all the data of the environment at one point.

But this is not enough, we design orbit of rotation to scan the environment which is similar to 360-degree scanning.

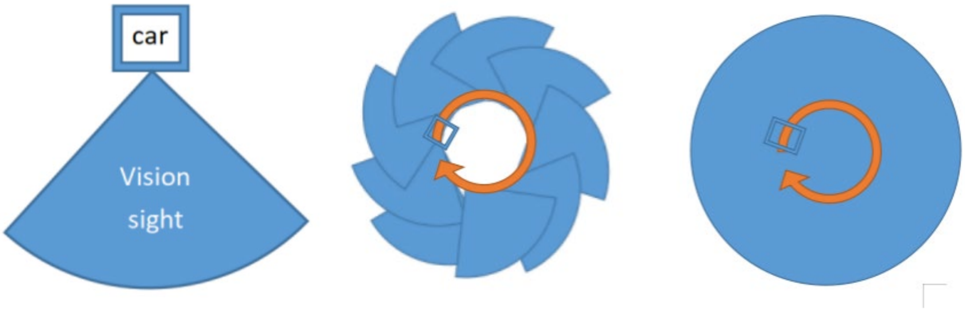


Figure 8: How the orbit helps the robot scan the environment in 360 degree

We want to completely scan the environment which includes different angles of the object. The method is shown in figure 9. So that it trains with more clearer features. As the graph shows at the begging the robot starts from the bottom of the triangle, robot starts to rotate and take pictures. This picture includes the right side of the box. Then turn 30 degrees clockwise and drive to the next location. Repeat the first step but take pictures of the left side of the box. Lastly turn 120 degrees anti-clockwise and go to the final location repeat the first step but take the pictures of wide range. The robot not only takes pictures of an object but also takes pictures of every object in the environment. (Orbit is shown below, blue line represents orbit black line represents vision sight.)

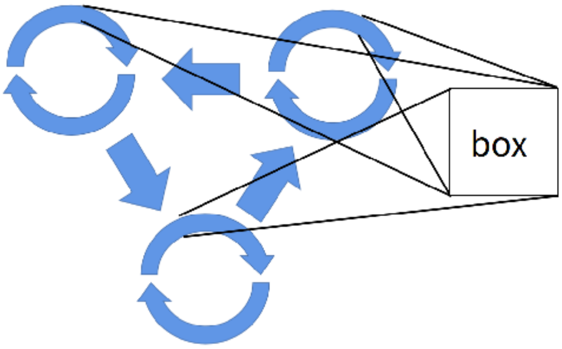


Figure 9: How the robot scan the box in different angle

The training of model requires pictures and annotation. Existing online reasoning services provide very limited recognition tasks, and pre-trained models are usually obtained from public datasets, and the prediction accuracy for specific scenarios is not high, so we can only label the data in other ways. We chose a more efficient method to label the images. We use detectron2 to bring out the location of two diagonally angle of the box which box out the object. When the detection has finished, we adjust the annotation by human eyes, and add new annotation when the object was unable to be recognized. Using this annotation, we create a coco json format file, which is able to train the model with the images.

3.2. Methodology

3.2.1. Detecton2 network architecture

Detectron2 is based on R-cnn. So, they use the same network architecture which is Feature Pyramid Networks. Different kinds of pyramid are shown in figure 10. Feature Pyramid Networks is widely used method of object detection. Different characteristics of objects in different pictures show different sizes, conclude different result of object detection. So, process image in a single scale is not enough for object detecting, the model combines the concept of pyramid, which means solving problems in different scale from the smallest level gradually increase to the biggest scale. This network structure makes results better and more comprehensive.

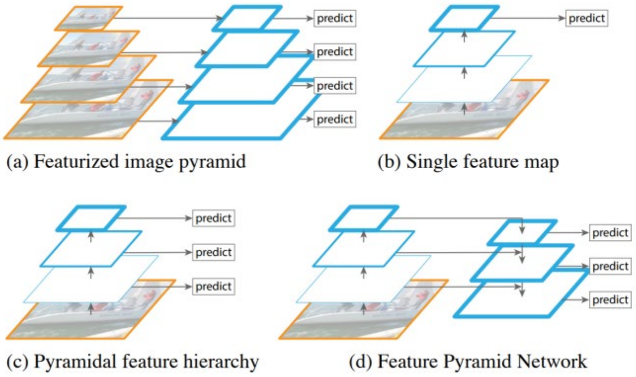
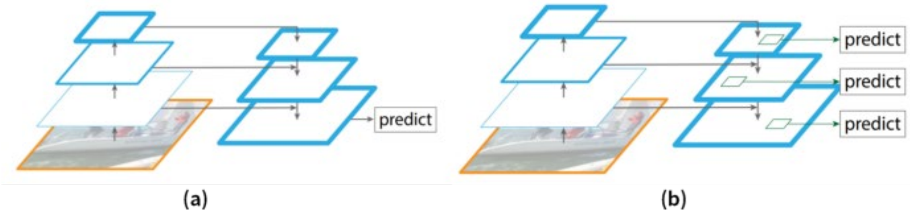


Figure 10: How Feature Pyramid Network evolved.

There are 4 pyramids which are shown above, blue boxes represent feature maps. The thickness of the frame of the blue box means the level of feature, thicker frame means the higher the level of the feature.

These 4 methods have a progressive relationship(a) feature image pyramid method is the earliest usage of pyramid, which simply extract different size of images then classify each size of the image. This method is widely used in the age of human annotation. But it repeats lots of unnecessary work and wastes a lot of time. Method shown in (b) single feature map using a different process, which model only predicts or classifies the images in the last feature map. Most of the famous object detector (CNN) are the base of this theory. Although this structure present faster result then (a) feature image pyramid, it is still not fast enough. (c) Pyramidal feature hierarchy using feature map produce by convoluted and increase level of size of image to predict. SSD network is the first network accomplish this structure, but SSD only starts to use the feature maps after the conv4_3 layer of the backbone VGG16. Although these feature maps have little meaning. It is very important for detecting small objects. This structure can avoid lots of unnecessary work, but there are still some flaws bad for detection. (d) Feature pyramid network (FPN) combine the concept of top-down and skip-connection, which makes every feature map meaningful. It ensures the accuracy and speed of detection. How the pyramid works is shown in figure 11.

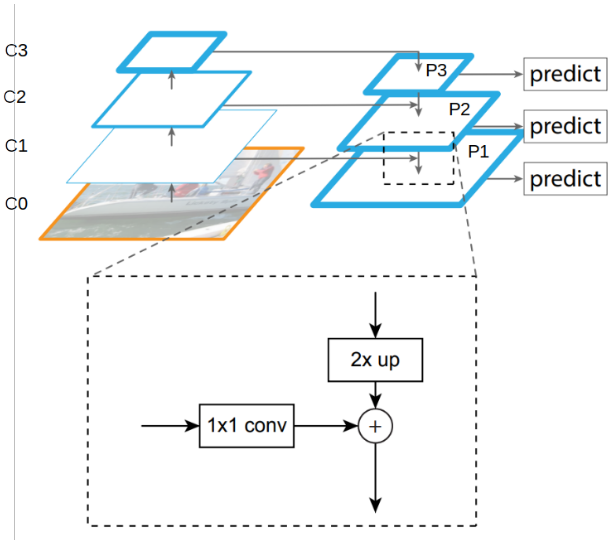


*https://miro.medium.com/max/1400/1*5_nIrywbiiSwxGHjSIvzrA.png*

Figure 11: How Pyramid Network functioned.

Detecron2 is using Feature Pyramid Network. The structure is shown above. High level feature map after upsampling add up the feature map of lower level, this design make feature in different size integrate. Also Feature Pyramid Network predicts in different sizes of feature maps instead of predicting the last feature map.

The process of upsampling is, first double the size of smaller feature map. Then it became the same size as the upper level. However, the upsampling here is a simple interpolation method, not through deconvolution. C1 will first go through 1x1 conv. Adjust the depth to be the same as P2, and then do element-wise addition with P2. After the addition, the 3x3 conv. will be used to eliminate the side effects of upsampling, it will be used as the final feature map to be used. How Pyramid Network calculates at each level is shown in figure 12.



*https://miro.medium.com/max/1400/1*HcBQmT5QNYKx764WNZuLwQ.png*

Figure 12: How Pyramid Network calculate at each level.

The strategy of Feature Pyramid Network (FPN) is using the equation below and Pk0 to calculate the height and width:

$$\mathrm{k} = \lfloor \mathrm{k}_0 + \log_2(\sqrt{\mathrm{wh}}/224) \rfloor$$

The design of Feature Pyramid Network (FPN) improves the way of connecting different size and level of feature maps. It makes the detection better[5].

3.2.2. Optimizing process

During the prediction, the model optimizes using this method. Hibbard using horizontal and vertical gradients computed each pixel, G component is going to estimate this pixel which is named gradient computation. To find the best fit green level, we need to consider CFA structure. Two steps to estimate the green level G.

1) Approximate the horizontal Δx and vertical Δy gradients using absolute difference.

$$\Delta^x = |G_{-1,0} - G_{1,0}|$$

$$\Delta^y = |G_{0,-1} - G_{0,1}|$$

2) Interpolate the green level.

$$\hat{G} = \begin{cases} (G_{-1,0} + G_{1,0})/2 & if\ \Delta^x < \Delta^y \\ (G_{0,-1} + G_{0,1})/2 & if\ \Delta^x > \Delta^y \\ (G_{-1,0} + G_{1,0} + G_{0,-1} + G_{0,1})/4 & if\ \Delta^x > \Delta^y \end{cases}$$

Stochastic gradient descent is also a widely used method of optimizing deep-learning algorithm. The maximum value of the upward Derivative represents the direction of gradient. During the process of decrease gradient, in order to find the best solution, it updates the importance in the opposite direction of the gradient. This process of update θi can be describe as:

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta)$$

$$\frac{\partial}{\partial \theta_j} J(\theta) = \frac{\partial}{\partial \theta_j} \frac{1}{2} (h_\theta(x) - y)^2$$

$$= 2 \cdot \frac{1}{2} (h_\theta(x) - y) \cdot \frac{\partial}{\partial \theta_j} (h_\theta(x) - y)$$

$$= (h_\theta(x) - y) \cdot \frac{\partial}{\partial \theta_j} (\sum_{i=0}^{n} \theta_i x_i - y)$$

$$= (h_\theta(x) - y)x_j$$

The third method to optimize is fine-tuning method,which is the method we are using. This method optimizes the model in four steps, which is shown in figure 13.

1) Pretrain a neural network model.

2) Create a new neural network model.

3) Add an output layer to the target model.

4) Train the target model on the target dataset.

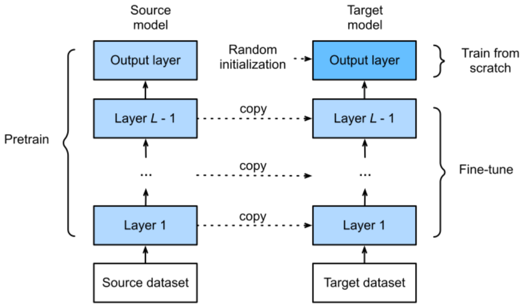


Figure 13: How fine-tuning method worked.

4. Experimental evaluation

4.1. Experimental Setup

We will train the open-source model "detectron2", using real world scenes. Detectron2 is for object detection. We want to make it especially for automatic driving models. So, we built a robot which can simulate driving in reality, which is equipped with a camera. So that the images taken are more similar to reality in vision sight and orbit. We choose JetBot which is an open-source robot based on NVIDIA jetson Nano because it's Educational. We use sd card to mirror writing the system which is download form NVIDIA official website. Now we have a car which takes photos.

4.2. Show actual computational results

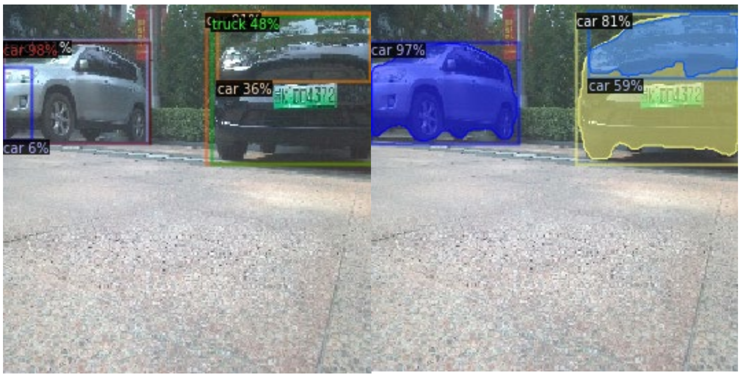


Figure 14: The prediction of object detection model.

On the left side of Figure 14 is the prediction of object detection model before training. On the right side of figure 14 is the prediction of object detection model after training. The figure indicates the success of our training.

4.3. Display diagram

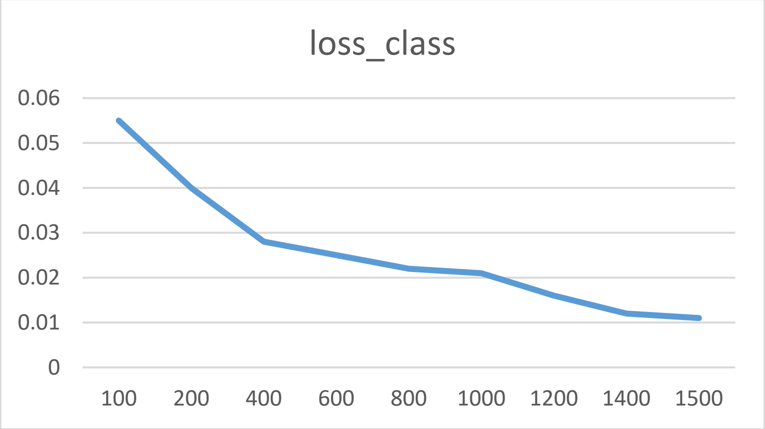


Figure 15: Changes of losses of class in training 1500 times.

Figure 15 describes the change of loss of classes, which is gradually decreased.

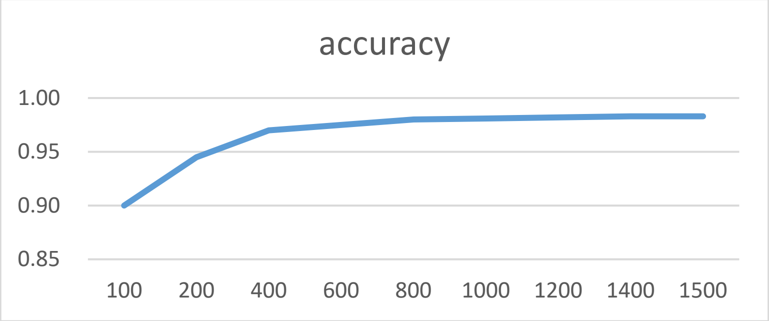


Figure 16: Changes of accuracy of class in training 1500 times.

Figure 16 shows that the accuracy of the model is gradually improved.

5. Conclusion

Object detection came into our life imperceptibly. The automatic drive is an important achievement of success in object detection. The leader of this industry is Tesla, which also had a lot of accidents as well. Object detection has a long way to go, and it takes time and evidence to make people believe in the automatic drive. Object detection can assist automatic driving and also use in daily life. We can use cameras to build alarm systems or other security equipment. To promote object detection, we need more people to work on optimizing the detection model.

References

[1] Tian Yuchi, Kexin Pei, Suman Jana, and Baishakhi Ray (2018). "Deeptest: Automated testing of deep-neural-network-driven autonomous cars." In Proceedings of the 40th international conference on software engineering, pp303-314.

[2] Zhao Zhongqiu, Peng Zheng, Shoutao Xu, and Xindong Wu (2019). "Object detection with deep learning: A review." IEEE transactions on neural networks and learning systems 30, no. 11: pp3212-3232.

[3] Slack, Dylan, Sophie Hilgard, Emily Jia, Sameer Singh, and Himabindu Lakkaraju(2020). "Fooling lime and shap: Adversarial attacks on post hoc explanation methods." In Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society, pp. 180-186.

[4] George Karimpanal, Thommen; Bouffanais, Roland (2019). "Self-organizing maps for storage and transfer of knowledge in reinforcement learning". Adaptive Behavior. 27 (2):pp111–126.

[5] Tsung-Yi Lin, Piotr Dollar, Ross Girshick, Kaiming He, Bharath Hariharan(2017). Serge Belongie; Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2117-2125.