Object Detection Based on YOLO Network

Chengji Liu[1] ,Yufan Tao[1] ,Jiawei Liang[1] ,Kai Li[1] ,Yihang Chen[1]
1. School of Mechanical Electronic and Information Engineering,
China University of Mining and Technology(Beijing), Beijing, China
{l_chengji, T_yufan787, zgkdljw, yihangc, LiBronKai}@163.com

***Abstract*—Object detection based on the deep learning has achieved very good performances. However, there are many problems with images in real-world shooting such as noise, blurring and rotating jitter, etc. These problems have an important impact on object detection. Using traffic signs as an example, we established image degradation models which are based on YOLO network and combined traditional image processing methods to simulate the problems existing in real-world shooting. After establishing the different degradation models, we compared the effects of different degradation models on object detection. We used the YOLO network to train a robust model to improve the average precision (AP) of traffic signs detection in real scenes.**

Keywords—YOLO network, image processing, object detection

I. INTRODUCTION

Object detection is one of the most important research directions for computer vision. Object detection algorithms are mainly divided into the traditional methods [1,2,3]and the deep learning methods [4,5,6]. In addition, the latter can be roughly divided into two categories. Some are based on region proposal object detection algorithms, including RCNN [7], SPP-net [8], Fast-RCNN [9] and Faster-RCNN [10], etc., which are algorithms that generate region proposal network and classify these region proposals afterwards. The others are based on the regression object detection algorithm, including SSD [11] and YOLO [12,13,14], etc. These algorithms generate region proposal network and classify these region proposals at the same time. All algorithms have good performances in object detection.

However, these algorithms are not tested with degraded images. In other words, they are trained with academic data sets, including ImageNet [15], COCO [16] and VOC [17], etc., but not well tested with randomly captured data sets. The main issues of images captured in the real scene include:

1) Due to the instability of the camera, the captured images can be blurred.

2) The images are not clear enough because the object can be obstructed.

3) The images may have poor quality as a result of bad weather, overexposure or low resolution. (see Fig. 1)

Fig. 1. Image problems: underexposure, rotation, blur, noise

The same object with small deviations in pixels caused by actual shooting may be divided into different classifications by neural networks, even though they are identical to human eyes [18].

In this paper, we simulated different degenerative processes of images for analysis and research. Firstly, we established the models of degraded images. We mainly used mathematical models to generate degraded images which are based on standard data sets. Then, we used these models to train the network to adapt to the complex real-world environment. Finally, we improved the ability of the model to generalize complex images. We took the traffic signs as the research object and used the YOLO [14] neural network to analyze. The experiment was based on the Darknet-53 [14] network structure. As a result, we made the following contributions:

1) We established a new image degradation model and used different degraded images as test sets. Then, we compared the effect of different degraded images on the standard model.

2) We modified the source network and performed different degradation processes for the training set. We also compared the accuracy of test sets on different models. Then, we performed more complex degradation processes on training sets and obtained a more generalized detection network. Afterwards, we compare it with 1) test performances.

3) Based on the above, we optimized the object detection method. To conclude, the generalization ability of the model has been enhanced, and the accuracy of object detection has been improved.

II. IMAGE DEGRADATION

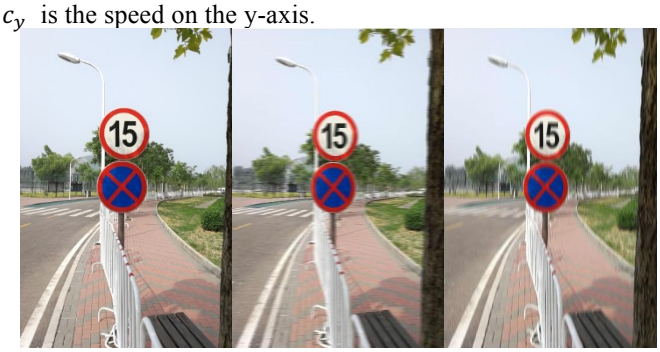
We made some rules for ease of description: First, we assumed the coordinates origin (0,0) to be the lower left corner of the image. Second, the images were all in the first quadrant. What's more, the x-axis was the width of the image and the y-axis was the height of the image.

978-1-5386-5373-9/18/$31.00 ©2018 IEEE

A. Blur [19, 20]: The representation of blur includes motion blur, disk blur and Gaussian blur, etc. In general, motion blur means the relative speed of the image is not changed at the moment of exposure because the shutter time is very short. Thus, the blur is mainly caused by the uniform linear motion of the object during this moment. It can be considered that the blurred image is caused by overlapping images (see Fig.2). We assumed the images without noise and blur are f (x, y), while g (x, y) stands for the blurred images. The formula was:

$$g(x,y) = \int_0^t f(x + c_x t, y + c_y t)\, dt \quad (1)$$

where t is the exposure time, c_x is the speed on the x-axis and c_y is the speed on the y-axis.



(a) (b) (c)

Fig. 2. (a) Original images (b) Images with blur along x-axis respectively (c) Images with blur along y-axis respectively

B. Rotate(Flip): Sometimes the camera has a certain tilt angle when shooting, so images will have a degree of rotation. Thus, we did random rotation of -10° to 10° angle around the center of the images. We assumed the width of image is x, the height of image is y and the angle of rotation is β**.** The formula was:

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = (x \quad y) \begin{pmatrix} cos\beta & sin\beta \\ -sin\beta & cos\beta \end{pmatrix} \quad (2)$$

After rotation transformation, x' became the width of new image and y' became the height of new image. (see Fig. 3)

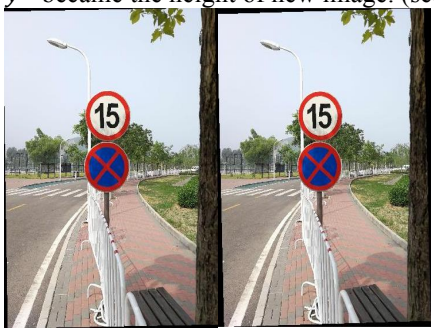


Fig. 3. Images with random rotation

C. Noise [19,21]: There may be a lot of noises existing in images due to some reasons such as electromagnetic interference and low illumination level, etc. Gaussian noise, salt & pepper noise and uniform noise are common noises in the images. We chose Gaussian noise and salt & pepper noise as representative noises for image degradation processes.

The formula of Gaussian noise was:

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) \quad (3)$$

where μ is the mathematical expectations and σ is the standard deviation.

The formula of salt & pepper noise was:

$$f(x) = \begin{cases} Pa & x = a \\ Pb & x = b \\ 1 - Pa - Pb, & otherwise \end{cases} \quad (4)$$

The value of salt & pepper noise in this experiment was 0.08. That means 8% pixels of the images would be randomly assigned 0 or 255. (see Fig. 4)

(a) (b)

Fig. 4 (a) Salt & pepper noise (b) Gaussian noise

***D. Cropping*:** When the subject is obstructed by other objects, we can only photograph a part of the object. Thus, we cropped the image randomly. Both cropping width and height vary from 0 to 0.15.

(a) (b)

Fig. 5. (a) Original images (b) Image with random cropping

III. YOLO NETWORK

The YOLO neural network integrates the candidate boxes extraction, feature extraction and objects classification methods into a neural network. The YOLO neural network directly extracts candidate boxes from images and objects are detected through the entire image features.

Traffic sign detection refers to applying the candidate box extraction technique to the input image to determine whether it contains traffic signs and output their location. The data set we used in this paper for training and evaluation comes from ImageNet. None of the images are tagged and they are independent from the ones used in pre-training. We selected 1,652 images containing traffic signs as the data set. Then, 1,318 images were selected which form the standard training set, and the remaining 334 images were used as the test set. All traffic signs in the images are labelled. Furthermore, we chose

334 images which were taken in the real world to form the test set that analyzed the model's performance. All these images have degeneration processing. The background of these images is complex, the light of the images has obviously changed and they have different degrees of occlusion.

We use the YOLOv3[14] neural network on an Intel CPU i7 -4790 16G memory, NVIDIA GTX1080, Ubuntu16.04, a 64-bit operating system for model training and testing. We use 0.9 momenta and 0.0005 decay in order to speed up training and prevent overfitting. The learning rate will adjust according to the training set. All models are based on the Darknet-53 neural network and use the same initial weight values for training, while it is different for the training sets.

In the YOLO network, the images are divided into S×S grids [11]. Candidate boxes are equally distributed on the X-axis and Y-axis. The candidate boxes have object detection and predict the confidence of the existence of the object in each candidate box. Confidence reflects whether the images include the object or not, as well as the accuracy of the object's position. We name confidence as Conf(object). The formula for Conf(object) is:

$$\mathrm{Conf}(\mathrm{Object}) = \Pr(\mathrm{Object}) \times \mathrm{IOU}_{\mathrm{Pred}}^{\mathrm{Truth}} \quad (5)$$

where $\Pr(\mathrm{object}) \in \{0,1\}$.

$$\mathrm{IOU}_{\mathrm{Pred}}^{\mathrm{Truth}} = \frac{area(box(Truth) \cap box(Pred))}{area(box(Truth) \cup box(Pred))} \quad (6)$$

$\mathrm{IOU}_{\mathrm{Pred}}^{\mathrm{Truth}}$ is the ratio of the prediction box to the truth box.

The conditional probability of the existence of traffic signs given the object detection is $\Pr(\mathrm{Traffic}|\mathrm{Object})$ and the confidence of a candidate box which includes the confidence in the existence of traffic signs is:

$$\begin{aligned} \mathrm{Conf} &= \Pr(\mathrm{Traffic}|\mathrm{Object}) \times \Pr(\mathrm{Object}) \times \mathrm{IOU}_{\mathrm{Pred}}^{\mathrm{Truth}} \\ &= \Pr(\mathrm{Traffic}) \times \mathrm{IOU}_{\mathrm{Pred}}^{\mathrm{Truth}} \end{aligned} \quad (7)$$

IV. Experiments and Results

A. The results from different test sets

We selected 80% of the data in the standard set as train sets. We do not perform image degradation on train sets; instead, feed the train set to the YOLO neural network to get the standard model (Ms). Then we use the remaining 20% of the original data for testing to obtain the test results of the standard model. Finally, we perform a single image degradation on the test sets, after which we feed the new test sets to the standard model (Ms). The test results of the degraded test sets in the standard model are as follow (see Table I).

Table I. Different Test Sets

Name	Test Sets	AP
Model Standard (Ms)	None	87.75
Model Standard (Ms)	Blur	85.69
Model Standard (Ms)	Noise	73.54
Model Standard (Ms)	Rotate(Flip)	63.52
Model Standard (Ms)	Cropping	59.79

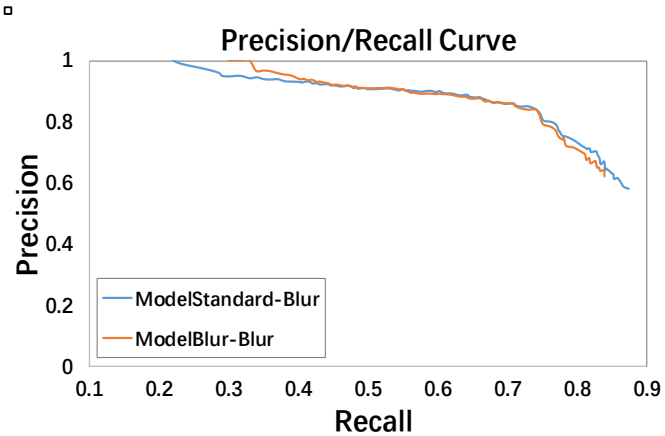
From table 1, we can observe that all image degradation processes make the average precision levels decrease. Then we find that the degradation processing of blur makes the average precision(AP) levels decrease by 2.06%; the degradation processing of noise makes the average precision(AP) levels decrease by 12.01%; the degradation processing of rotation(Flip) makes the average precision(AP) levels decrease by 24.23%; the degradation processing of cropping makes the average precision(AP) levels decrease by 27.96%. Obviously, the rotate(Flip) and cropping transformation have the greatest impact on model performance. The model trained with the standard sets does not have good generalization ability for the degraded images and has poor robustness. We believe the reasons are: (1) the model which is trained by the standard set is overfitting; (2) the test sets degenerate from the statistical distribution of the training set after image degradation processing. These reasons may cause the fall of average precision(AP).

B. The results with different models

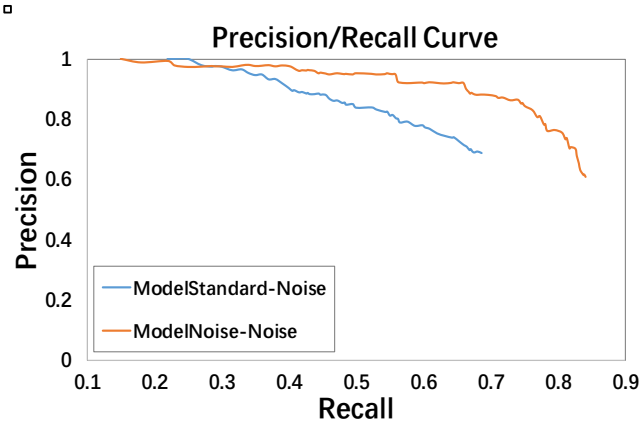
We select 80% of the data in the standard set as the training set. Then we perform a single image degradation processing on the images in the training sets. As a result, we obtain the corresponding models (Mm, Mn, etc.). Finally we used the remaining 20% data to test and obtain the test performance of the corresponding models. (see Table II and Fig. 6)

Table II. Different Models

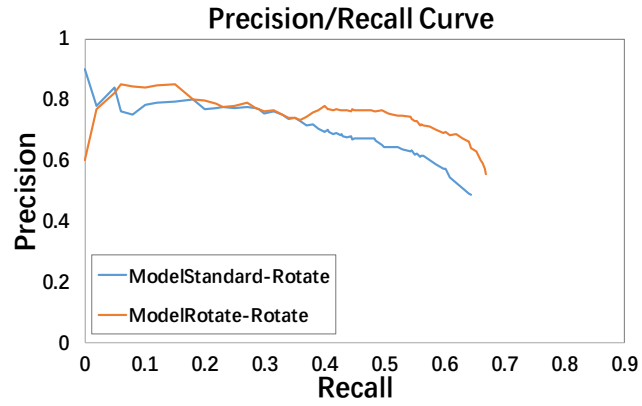
Name	Training sets	AP
Model Standard (Ms)	Blur	85.39
Model Blur(Mm)	Blur	86.93
Model Standard (Ms)	Noise	73.54
Model Noise (Mn)	Noise	84.63
Model Standard (Ms)	Rotate(Flip)	63.52
Model Rotation (Mr)	Rotate(Flip)	64.46
Model Standard (Ms)	Crop	59.79
Model Crop (Mc)	Crop	68.03



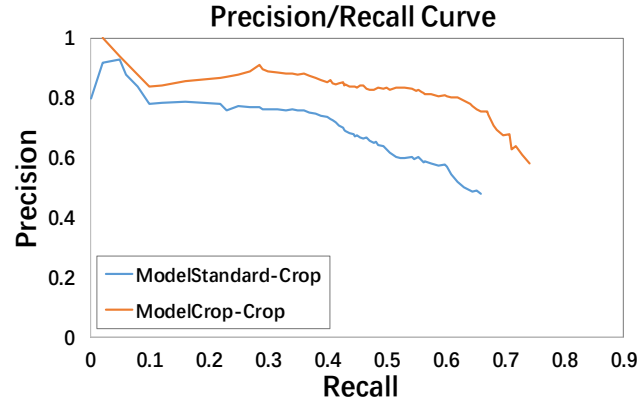
(a). AP of different models(Blur)



(b). AP of different models(Noise)



(c). AP of different models(Rotate)



(d). AP of different models(Crop)

Fig 7. (a)(b)(c)(d) Precision/Recall of different models

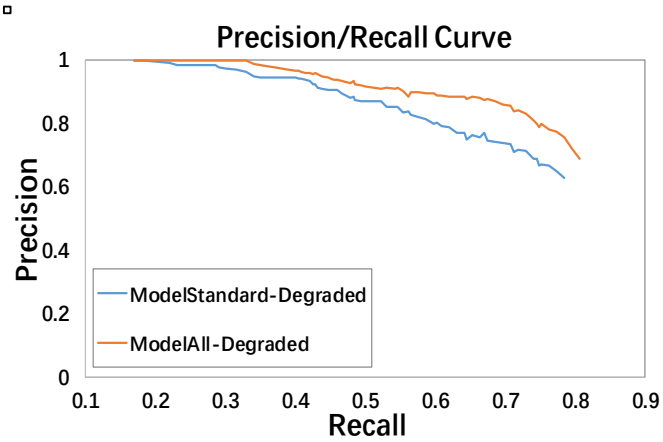
From table 2 and Fig. 7, The training model after degradation processing has different degrees of improvement to the average precision(AP) of the test set. We can observe that all image degradation processes determine the average precision levels to improve. We find that the degradation processing of blur makes the average precision (AP) levels increase by 1.54%; the degradation processing of noise makes the average precision (AP) levels increase by 11.09%; the degradation processing of rotation (Flip) makes the average precision (AP) levels increase by 0.93%; the degradation processing of cropping makes the average precision (AP) levels increase by 8.24%. It is obvious that the degraded training set can improve the generalization ability and robustness of the model.

C. The result about general degenerative models

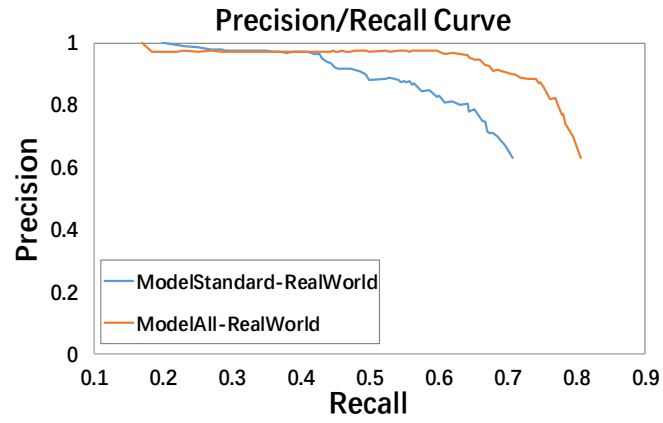
We evaluate the performance of a single degenerate model. Next, we select 80% of the data in the standard set which contains all images that went through the general degradation processing. We feed these images to the network to train the general degenerative model(Ma), while the remaining 20% of data are used for testing. Finally, we compare the general degenerative model with the standard model. (see Table III and Fig. 8)

Table III. General degenerative model

Name	Training set	AP
Model Standard (Ms)	Degraded	65.72
Model Standard (Ms)	RealWorld	83.46
Model All (Ma)	Degraded	73.22
Model All (Ma)	RealWorld	85.78



(a). AP of degenerative models(Degraded)



(b).AP of degenerative models(RealWorld)

Fig. 8.(a)(b) Precision of general degenerative models

We find the average precision(AP) of the general degenerative model(Ma) is better than the standard model of all test sets. We believe that the network which is trained with

degraded images learns more features and can cope with more complex scenes. The model has better generalization ability and higher robustness.

V. CONCLUSION

In this paper, we propose using the YOLO network model for object detection. We train the degenerative model which is fed with the degraded image. Through experiments we find that the network which is trained with the degraded images learns more features and that the model can cope with more complex scenes. The results show that the model improves the average precision of the object detection. The model which is trained with the degraded training sets has better generalizing ability and higher robustness.

REFERENCES

[1] Lowe D G. Object recognition from local scale-invariant features. International Conference on Computer Vision, 1999: 1150-1157.

[2] Hearst M A, Dumais S T, Osman E, et al. Support vector machines. IEEE Intelligent Systems &Their Applications, 1998,13(4):18-28.

[3] Freund Y, Schapire R E. A decision-theoretic generalization of on-line learning and an application to boosting. Computational Learning Theory, 1995:23-27.

[4] Dollar P, Tu Z, Perona P, et al. Integral Channel Features. British Machine Vision Conference, BMVC, 2009:1-11.

[5] Dollar P, Appel R, Belongie S, et al. Fast Feature Pyramids for Object Detection. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2014, 36(8):1532-45.

[6] Druzhkov P N, Kustikova V D. A survey of deep learning methods and software tools for image classification and object detection. Pattern Recognition and Image Analysis, 2016, 26(1):9-15.

[7] Girshick R, Donahue J, Darrell T, et al. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. Computer Vision and Pattern Recognition, 2014:580-587.

[8] He K, Zhang X, Ren S, et al. Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. European Conference on Computer Vision. Springer, Cham, 2014:346-361.

[9] Girshick R. Fast R-CNN. IEEE International Conference on Computer Vision. IEEE,2015:1440-1448.

[10]]Ren S, He K, Girshick R, et al. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2017, 39(6):1137-1149.

[11] L. Wei, A. Dragomir: SSD: Single Shot MultiBox Detector. arXiv:1512.02325v5,2016.

[12] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. arXiv:1506.02640, 2015

[13] Redmon J, Farhadi A. YOLO9000: Better, Faster, Stronger. arXiv:1612.08242v1,2016

[14] J.Redmom A.Farhadi. YOLOv3: An Incremental Improvement arXiv:1804.02640,2018

[15] Deng J, Dong W, Socher R, et al. ImageNet: A large-scale hierarchical image database. Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on. IEEE, 2009:248-255.

[16] Lin T Y, Maire M, Belongie S, et al. Microsoft COCO: Common Objects in Context. arXiv:1405.0312,2014.

[17] Everingham M, Gool L, Williams C K, et al. The Pascal Visual Object Classes (VOC) Challenge. International Journal of Computer Vision, 2010, 88(2):303-338.

[18] Paulin M, Revaud J, Harchaoui Z, et al. Transformation Pursuit for Image Classification. Computer Vision and Pattern Recognition. IEEE, 2014:3646-3653.

[19] Gonzalez R, Woods R. DIGITAL IMAGE PROCESSING. Publishing House of Electronics Industry, 2010

[20] Sorel M, Flusser J. Space-Variant Restoration of Images Degraded by Camera Motion Blur. IEEE Transactions on Image Processing A Publication of the IEEE Signal Processing Society, 2008, 17(2):105-116.

[21] Xing-Mei LI, Chen L. The Wavelet Threshold Deleting Noise Method in Image Nosie Deletion. Modern Computer, 2006.