International Journal of Database Theory and Application
Vol.9, No.6 (2016), pp.45-52
http://dx.doi.org/10.14257/ijdta.2016.9.6.05

A Novel Dataset Generating Method for Fine-Grained Vehicle Classification with CNN

Shaoyong Yu[1,2] ,Zhijun Song[3] ,Songzhi Su[1] ,Wei Li[2] ,Yun Wu[2] and Wenhua Zeng[1*]

[1]Department of Cognitive Science, Xiamen University, Xiamen 361000, China
[2] School of Computer and Information Engineering, Xiamen University of Technology, Xiamen 361024, China
[3] The 28th Research Institute of China Electronics Technology Group Corporation, Nanjing 210007, China
syyu@xmut.edu.cn

Abstract

We focus on the issue of dataset generation for fine-grained vehicle classification with CNN. Traditionally, to build a large dataset, images must be first collected manually, and then be annotated with a lot of effort. All these work are time-consuming and cost-prohibitive. In this work we propose a novel method that can generate massive images automatically, and these generated images need no annotation. An AutoCAD 3D model of a car of specified make and model is imported into our system, and then images of different views of the car are generated, these images can describe all the details of a car. By taking these images as training dataset, we use a Convolutional Neural Network to train a model for fine-grained vehicle classification. Experimental results show that these images generated virtually by 3D model indeed work as effective as real images.

Keywords*: Dataset Generation; Fine-Grained; Vehicle Classification; CNN; 3D*

1. Introduction

With the availability of large scale training dataset, deep convolutional neural networks based approaches have recently been substantially improving upon the state of the art in image classification [1-4],object detection [3,5-6], and many other recognitions tasks[7-10]. But in early times, lack of datasets and limited computation ability of CPU/GPU restrict CNN only to be applied in small problem domain like digital hand written digit recognition. So we can conclude that training dataset is essential to CNN model.

There are a lot of datasets publicly available now, from small scale to large scale. Small image datasets like Caltech101/256 [11-12], MSRC [13], PASCAL [14] have served as training and evaluation benchmarks for most of today's computer vision algorithms. As computer vision research go further, larger datasets are needed. So, datasets like TinyImage [15] which has 80 million images, all these images are acquired from image search engines like Google,Baidu,Bing and so on by using keywords. Other larger datasets like LableMe [16], Lotus Hill [17] and ImageNet [18] provide 30k, 50k and 50 million labeled and segmented images respectively, which need massive people to annotate.

Despite of large quantity of images, it is still not enough for deep learning model. Neural network architecture usually has millions of parameters, existed datasets turn out to be insufficient to learn so many parameters without considerable overfitting.So Researchers take some technical methods like cropping [19], resizing [1,20], mirror reflection [21-22] to augment the existed datasets.

ISSN: 2005-4270 IJDTA

So we can jump to the conclusion that to build a large scale dataset, one should first collect massive images by internet search engines, and then employ lots of people to annotate them, after that technical methods are used to augment the datasets. Even though, images in the datasets cannot cover all the views of a specified type object.

Can we have an easier and faster way to do this? So we propose a novel method that uses a 3D model to generate 2D images of all views. By using this way, we have no need to gather images from the image search engines; still we do not need to employ people to annotate these images.

2. Dataset Generating Method

For fine-grained vehicle detection, we use an AutoCAD 3D model of a car of specified make and model. By changing the camera distance, direction angle and over angle we can get tens of thousands of images of car with different appearance. This sort of images needs no annotation, which can save our effort.

First, 3D car model was put on the place where car floor's center point coincide with the original point. Then we use three parameters to adjust the camera view, which can generate different 2D image. These three parameters are distance, direction angle β and over angle α, where β ranges from 0 degree to 360 degree, α changes between 0 degree to 90 degree because of symmetry. So we can get the camera coordinate as follows, refer to Figure 1.

$$\begin{aligned} x &= dis\tan ce * \cos(\alpha / 180.0 * \pi) * \sin\ (\beta / 180.0 * \pi) \\ y &= dis\tan ce * \cos(\alpha / 180.0 * \pi) * \cos\ (\beta / 180.0 * \pi) \\ z &= dis\tan ce * \sin\ (\alpha / 180.0 * \pi) \end{aligned} \tag{1}$$

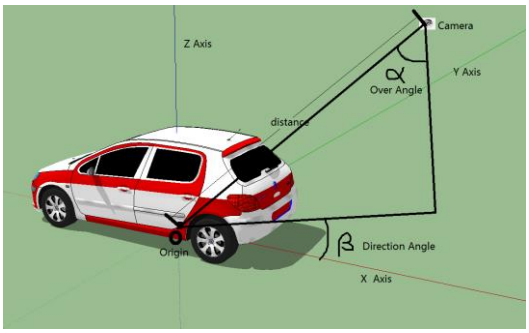

Figure 1. The Principle of Generating Car Images of All Views

And then we get massive images of this car model by the following steps as Figure 2 shows.

We import a 3D car model, and then set Camera distance as constant D, which decrease at a step of stepD, set over angle α and direction angle β as 0, which increase at a step of step α and step β respectively. In each loop, when D>0, α <90 and β <360, we can get from Equation 1 an coordinate, make it camera's position, and then output a 2D image of this car model. To be easily distinguished, we name the image file in the form of "D_ α _ β .jpg". Some images generated by this algorithm are as figure 3

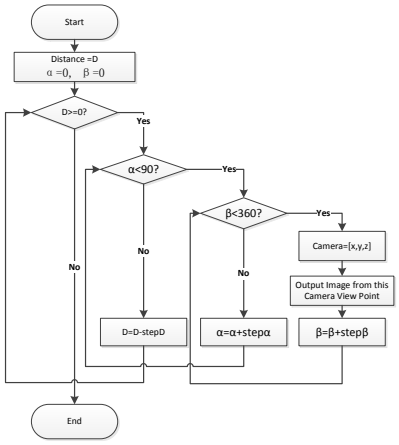

Figure 2. Algorithm Process of Generating 2D Images from 3D Model

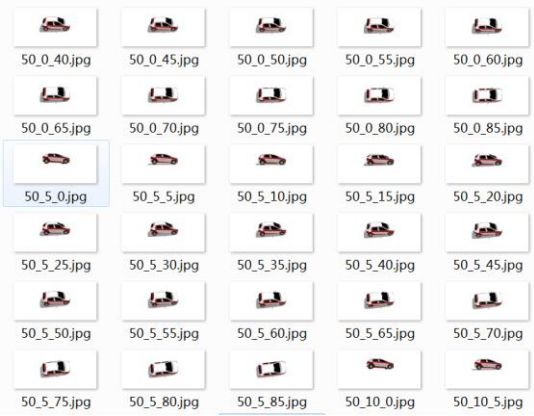

Figure 3. Images Generated by 3D Model

3. Experimental Results

Using the above dataset generating method, we can get massive images of a designated 3D car model. The following thing is to prove that this dataset is useful.

So I want to design a CNN model which can tell whether an image contains a car of the same model as 3D car model. In our experiments, we use a Peugeot 307 car 3D model.

By intuition, human can remember an object if provide all views of this object, the more views human see, the easier he can tell the object from others. Theoretically, if we feed this generated dataset into CNN model, that means we let the CNN model see every details of this object, so the model can easily distinguish it from other objects.

3.1. CNN Architecture

Now we are ready to describe our CNN architecture. As depicted in Figure 4, this net contains eight layers in total which the first five are convolution layers and the remaining three are fully-connected (FC) layers. The output of the last FC layer is fed to a 2-way softmax layer which can tell whether an image contains a Peugeot 307 car.

The kernels of the second, fourth, and fifth convolutional layers are connected only to those kernel maps in the previous layer which reside on the same GPU [20].The kernels of the third convolutional layer are connected to all kernel maps in

the second layer. The neurons in fc layers are connected to all neurons in the previous layer. Response-normalization layers follow the first and second convolutional layers. Max-pooling layers follow both response-normalization layers as well as the fifth convolutional layer. The ReLU non-linearity is applied to the output of every convolutional and FC layer.

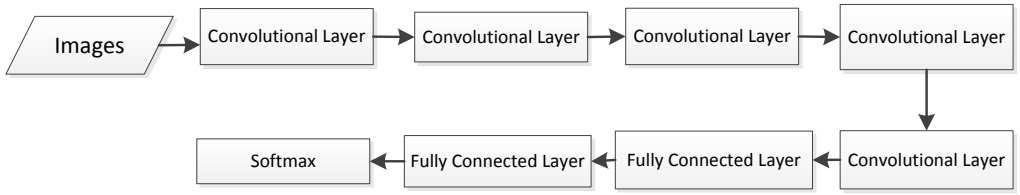

Figure 4. Overall CNN Architecture

3.2. Experiment Design

To verify that our virtually generated images is as useful as real images, we collect 7600 real images from image search engines, including 3800 images that contain Peugeot 307 and 3800 non-peugeot 307. By changing the over angle at a step of 3 degree, direction angle at a step of 5 degree, we use two different distance to generate (90/3)*(360/5)*2=4320 virtual images.

As real images that contain Peugeot 307, we separate it into two parts, 3000 images as positive training data, and 800 images as positive test data. We also separate images that contain no Peugeot 307 into two parts as Peugeot 307 images. All 4320 virtual images as treated as positive training data.

Then we perform the following six groups of experiments as Table 1.

Table 1. Experiment Design

	Real images (Positive Training)	Real images (Negative Training)	Virtual images (Positive Training)	Real images (Test)
Exp1	3000	3000	0	1600
Exp2	3000	3000	1000	1600
Exp3	3000	3000	2000	1600
Exp4	3000	3000	3000	1600
Exp5	3000	3000	4000	1600
Exp6	0	3000	4320	1600

In the first group of experiment, we only use real images, 3000 images as positive training data and 3000 images as negative training data and 1600 images as test data. From experiment 2 to experiment 5, we increase the number of virtual images in training data by 1000. In the last experiment, we use no real images for positive training data, only virtual images are included.

These experiments can be divided into three groups, that is real data with CNN, virtual data with CNN and combined data with CNN.

3.3. Discussion

As we can see in Table 2, pure real data can get a result of accuracy at 0.914, when 1000 positive virtual images were put into the training data, the accuracy becomes 0.935, that means virtual images is helpful. With the increasing join of virtual data, the accuracy becomes higher and higher, from 0.935 to 0.955. So we can conclude that the more virtual images, the higher the accuracy.

Table 2. Experimental Results

	Test Score 0# (Accuracy)	Test Score 1# (Loss)	Type
Exp1	0.914	1.4645244	Real data with CNN
Exp2	0.935	1.1342332	Combined data with CNN
Exp3	0.943	1.0125454	
Exp4	0.950	0.9985454	
Exp5	0.952	0.9854541	
Exp6	0.903	1.6562545	Virtual data with CNN

But we also notice that from Exp4 to Exp5, only 0.002 accuracy improvements, compared to previous 0.021, 0.008 and 0.007, why? By investing the generated images, we find that most of the images look similar, only with very a little difference. So we inferred that the last joined 1000 virtual images have nearly the same in the previously joined 3000 virtual images.

Let's see Exp6's result, you will find that the accuracy is lower than any of the other experiments. It looks very strange because virtual data in this experiment covers all the details of Peugeot 307 of all views; theoretically, the accuracy should be higher than Exp1.But why? After analysis, we get the idea. Also this the problem of 3D model, our 3D model only contains Peugeot 307 without any other background objects, but in test dataset are real images from internet, they are more complicated, nearly all images with clutter background. So we use software to only keep car in the test images and swipe out all other non-car portion manually, and then test again, the experimental result testify our guess. The result is as Table 3.

Table 3. Improved Experimental Results

	Test Score 0# (Accuracy)	Test Score 1# (Loss)	Type
Exp1	0.912	1.4542145	Real data with CNN
Exp2	0.937	1.1451235	Combined data with CNN
Exp3	0.948	1.0456254	
Exp4	0.955	0.9564455	
Exp5	0.959	0.9456855	
Exp6	0.925	1.2545545	Virtual data with CNN

When putting original and improved experimental results together, we can see that after preprocessing positive test data, a higher accuracy has been achieved. See Figure 5.

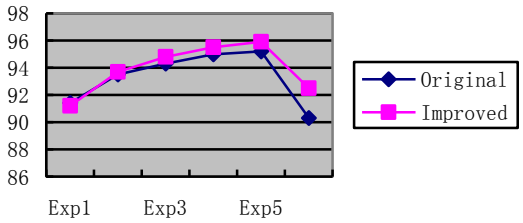

Figure 5. Accuracy Comparison of Original and Improved Experiments

4. Conclusion

As experimental results show, dataset generated virtually by 3D model can be used to train CNN classification models, and this kind of models can be utilized to classify real vehicles. The above experiments are just used to judge whether or not a real car image is a Peugeot 307 car, but if we want to tell between different vehicles makes and models, how to do this? Maybe a cascading CNN models can achieve this goal.

Another problem is that, because of our 3D model contains only clean car without any background, all the images generated are also very clean, but in fact in real car images which contain lots of non-car objects are not that clean. In our improved experiments, we manually preprocess the positive test data, but it is time-consuming. How to solve this condition to make the CNN model more adaptive? A probable way is to build a 3D car model with background objects like trees, roads, buildings and so on, so the generated virtual images will also contain complicated background.

Acknowledgments

This work is supported by Natural Science Foundation of Fujian Province of China (Grant No. 2013J05103 and No. 2015J05015 and No. 2016J01325) and High-level Personnel of Support Program of Xiamen University of Technology (Grant NO. YKJ14014R).

References

[1] K. Alex, I. Sutskever and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks", Advances in Neural Information Processing Systems, California, USA, **(2012)**.
[2] M. D. Zeiler and R. Fergus, "Visualizing and Understanding Convolutional Networks", ECCV 2014, Zurich, Switzerland, **(2014)**.
[3] S. Pierre, "OverFeat: Integrated Recognition, Localization and Detection using Convolutional Networks", Eprint Arxiv, **(2013)**.
[4] A.V.K. Chatfield, K. Simonyan and A. Zisserman, "Return of the devil in the details: Delving deep into convolutional nets", in ArXiv: 1405.3531, **(2014)**.
[5] R. Girshick, J. Donahue and T. Darrell, "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation", Computer Vision and Pattern Recognition, Columbus, Ohio, **(2014)**.
[6] W. Y. Zou, X. Wang and M. Sun, "Generic Object Detection with Dense Neural Patterns and Regionlets", Eprint Arxiv, **(2014)**.
[7] R. A. Sharif, "CNN Features Off-the-Shelf: An Astounding Baseline for Recognition", Computer Vision and Pattern Recognition, Columbus, Ohio, **(2014)**.
[8] Y. Taigman, M. Yang and M. Ranzato, "DeepFace: Closing the Gap to Human-Level Performance in Face Verification", Computer Vision and Pattern Recognition, Columbus, Ohio, **(2014)**.
[9] N. Zhang, M. Paluri and M. Ranzato, "PANDA: Pose Aligned Networks for Deep Attribute Modeling", Computer Vision and Pattern Recognition, Columbus, Ohio, **(2014)**.
[10] Y. Gong, L. Wang and R. Guo, "Multi-scale Orderless Pooling of Deep Convolutional Activation Features", Lecture Notes in Computer Science, vol. 3, no. 2, **(2014)**, pp. 392-407.
[11] L. F. Fei, R. Fergus and P. Perona, "One-shot learning of object categories", PAMI, vol. 28, no. 4, **(2006)**, pp. 594–611.

[12] G. Griffin, A. Holub and P. Perona, "Caltech-256 object category dataset", Technical Report 7694, Caltech, **(2007)**.
[13] J. Shotton, J. Winn, C. Rother, and A. Criminisi, "Textonboost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation", ECCV, Graz, Austria, **(2006)**.
[14] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL Visual Object Classes Challenge 2008 Results", http://www.pascal-network.org/challenges/VOC/voc2008/workshop/, **(2008)**.
[15] A. Torralba, R. Fergus and W. Freeman, "80 million tiny images: A large data set for nonparametric object and scene recognition", PAMI, vol. 30, no. 11, **(2008)**, pp. 1958-1970.
[16] B. Russell, A. Torralba, K. Murphy and W. Freeman, "Labelme: A database and web-based tool for image annotation", IJCV, vol. 77, no. 3, **(2008)**, pp. 157-173.
[17] B. Yao, X. Yang, and S. Zhu, "Introduction to a large-scale general purpose ground truth database: Methodology, annotation tool and benchmarks", CVPR, Minneapolis, USA, **(2007)**.
[18] J. Deng, W. Dong, R. Socher, L. J. Li, K. Li and L. F. Fei, "ImageNet: A large-scale hierarchical image database", CVPR, Florida, USA, **(2009)**.
[19] H. Kaiming, "Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition", IEEE Transactions on Pattern Analysis & Machine Intelligence", vol. 37, no. 9, **(2015)**, pp. 1904-1916.
[20] C. Jurgen, "Multi-column deep neural networks for image classification", CVPR, Providence, Rhode Island, **(2012)**.
[21] J. Yu, Y. Rui and D. Tao, "Click Prediction for Web Image Reranking using Multimodal Sparse Coding", IEEE Transactions on Image Processing, vol. 23, no. 5, **(2014)**, pp. 2019-2032.
[22] J. Yu, Y. Rui, Y. Tang and D. Tao, "High order Distance based Multiview Stochastic Learning in Image Classification", IEEE Transactions on Cybernetics, vol. 44, no. 12, **(2014)**, pp. 2431-2442.

Authors

Shaoyong Yu, a PhD student in Xiamen University, also a college teacher in School of Computer and Information Engineering at Xiamen University of Technology. His research interests lie in the areas of computer vision and deep learning. contact him at syyu@xmut.edu.cn.

Zhijun Song, Male, PhD. He received his PhD from Xiamen University in 2013. His research interests lie in the areas of artificial intelligence and big data. His scientific contribution to the AI has more to do with machine consciousness and the logic of mental self-reflection.

Songzhi Su, is an associate professor in the Department of Cognitive Science at Xiamen University. His research interests include computer vision, machine learning and its application, face recognition and pedestrian detection. He has a Ph.D. in Basic Theory of Artificial Intelligence from Xiamen University. Contact him at ssz@xmu.edu.cn.

Wei Li, is an associate professor in the School of Computer and Information Engineering at Xiamen University of Technology. His research interests include artificial intelligence, computer graphics. He has a Ph.D. in Basic Theory of Artificial Intelligence from Xiamen University. Contact him at weili@xmut.edu.cn.

Yun Wu, female, PhD She received her PhD from Xiamen University in 2007. Her research interests lie in the areas of artificial intelligence and big data. His scientific contribution to the AI has more to do with soft computing and the clustering algorithms.

Wenhua Zeng, is a professor in the Department of Cognitive Science at Xiamen University. His research interests include neural network, grid computing and embedded system. He has a Ph.D. in Industry Automation from Zhejiang University in 1986. Contact him at whzeng@xmu.edu.cn.