A Solution for Input Limit in CNN Due to Fully-Connected Layer

Yili Qu and Yaobin Ke
School of Data and Computer Science, Sun Yat-sen University
Guangzhou, Guangdong Province, China
{quyli & keyaobin}@mail2.sysu.edu.cn

Wei Yu
College of Computer, National University of Defense Technology
Changsha, Hunan Province, China
yuweimomo@163.com

Abstract—**Many CNNs require a fixed-size input due to the inclusion of fully-connected layer. In actual training and application, the size of input maybe various. The usual solution is cropping or warping. But cropping may lose useful pixels while warping will destroy the structural information thus causing geometric distortion potentially. Since there is still a wide range of application scenarios for fully-connected layer, we hope to solve this problem under the premise of retaining the fully-connected layer. In this paper, a solution based on the SPP improvement operator for CNNs containing fully-connected layer is proposed to help networks accept various size inputs and learn effectively. We build a dataset consisted of various-size geometry images, which is sensitive to geometric distortion. By the comparison of experiments on this dataset using different method, we verify that our method can significantly improve the prediction accuracy by avoiding the geometric distortion. By applying our method on AlexNet and VGGNet, we show the robustness of our method for models. On the public dataset VOC 2007 with unfixed image sizes and ImageNet with fixed image sizes, our solution can maintain the accuracy of the original model, indicating that our method is robust on ordinary datasets that are not sensitive to geometric distortion.**

Keywords—variable input, geometric distortion, SPP, pooling

I. INTRODUCTION

In general, the images we can get have various sizes. These large quantities of images have facilitated the development of computer vision technology based on convolution neural network (CNN) [1] [2]. CNN-based computer vision technology has played an important role in image classification [3][4][5][6], target detection [7][8][9] and many other computer vision tasks [10] [11] [12] [13].

At present, many CNNs are composed of two parts, the convolution part and the subsequent fully-connected part [14], which makes most CNNs currently have a limit in training and prediction. They require all input data to be the same size. The number of parameters in convolutional part is independent of the input size so a fixed size is not necessary. It can receive an input of any size to produce a feature map with a size related to the input size. On the other hand, the number of parameters of the fully-connected layer depends directly on the number of inputs and outputs of the neurons. The number of neurons in the last output layer is the same as the number of the classes, which is a fixed value. Only when the number of parameters of a network is fixed can the parameters be learned by iterative update. Due to the existence of the fully-connected layer, many CNNs require fixed-size input [14].

For images with different sizes, the common solution is to crop [3] [4] or warp [7] [15] the image to force all images into one uniform size, and then carry out CNN training and learning. However, many pixels with useful information are lost during the cropping process, which will influence the training accuracy. Cropping is not applicable in many scenarios [14]. Warping distortion will destroy the structural information such as the angle and proportion of the object, causing geometric distortion, which makes it difficult to learn the knowledge on many scientific data with geometric labels or high fidelity requirements. In the learning tasks of many natural image datasets, warping may not bring too much interference. However, with the development of computer vision, more and more scientific data are processed by CNN, which means many application scenarios will be sensitive to warp, and the preprocessing of images by warping will not meet the demand.

In 2014, Global Average Pooling(GAP) [16] was proposed. GAP can be used to replace the fully-connected layer of the model to avoid the fixed-size limit in input. Although GAP has achieved good results in some tasks, the fully-connected layer still plays a wide and important role in many computer vision tasks. Meanwhile, SPPNet [14] was proposed, the key strategy of which is called Space Pyramid Pooling (SPP) [17] [18]. This method can eliminate the fixed-size input limit. Regardless of the size or proportion of the input image, SPP can produce a fixed size output. SPPNet has achieved excellent results in ILSVRC 2014 [19] on the basis of retaining the fully-connected layer [14]. In view of the wide application scenarios of the fully-connected layer, we hope to solve the problems above using a better solution without abandoning fully-connected layer. SPP is currently the best solution for our problem. Unfortunately, previous studies do not conduct experiments and research on deformation problems. On the basis of them, we have studied the data deformation problem and found that SPP still brings geometric distortion.

We build a geometry image dataset[1] that is sensitive to warp and can effectively verify whether the model produces geometric distortion during preprocessing or training. Then we propose two SPP improvements, Variable Step Pooling(VSP)

[1]https://github.com/quyili/geometry_image_dataset

and Variable Step Convolution(VSC), which can be used more flexibly in tasks such as image classification, object detection, semantic segmentation, image generation, and super-resolution.

Base on VSP, we propose a solution for CNNs containing fully-connected layer to accept various size inputs. Firstly, it locates the effective information area of the feature map, which is the output of the last convolution layers in original network. Then it obtains a fixed-size output of the effective information area through the VSP. Finally it inputs to the fully-connected layer of the original network to predict. This solution enables CNNs with fully-connected layers that accept only fixed-size inputs to accept unfixed-size inputs and learn effectively. Compared with the ordinary CNN using the warping method and the SPP method, our method can significantly improve the prediction accuracy of the geometric-distortion-sensitive dataset. Applied on AlexNet and VGGNet, our method shows robust in the model. On the public dataset VOC 2007 [20] whose image size is fixed and ImageNet dataset whose image size is unfixed, our method can maintain the accuracy of the original model, indicating that our method is still robust on ordinary datasets that are not sensitive to geometric distortion.

II. GEOMETRY IMAGE DATASET

In some scientific data processing tasks, it may be necessary to make predictions based on the geometric characteristics of the data. As shown in Figure 1, taking the classification of biological cell as an example, when the biologists classify the cells by the figure, the geometry of the cells is an extremely important basis for judging [21]. In many cases, the size of the sample we can get is different. This is a kind of application scenario for us. But building such a dataset is very difficult currently.

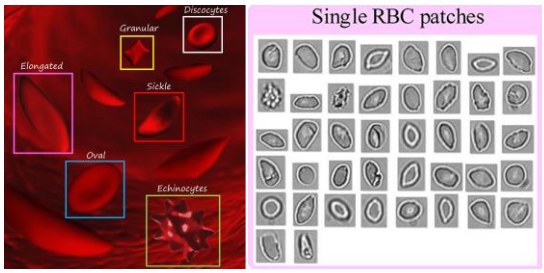

Figure 1. Different states of red blood cells with different shape (left) [22]; red blood cell images with various size (right) [21] .

Therefore, we hope to find a dataset that can simulate this situation. Unfortunately, there is no such a dataset whose labels are related to the geometric information of the picture and images have different size. Since it is so, we abstract the task and construct a geometry image dataset with an unfixed-size, including circular, elliptical, square, and rectangular. Each class has 1000 images, each of which has a length and width of any value from 256 to 1024. The size, color and position of the geometric block in the image are arbitrary in the appropriate interval. At the same time, we constructed another fixed-size dataset like unfixed-size geometry image dataset, except that the image size is fixed at 256×256. These datasets can simulate the above scenario very well, and can be easily constructed in Python. We randomly take 100 pictures for each class as the test dataset, and the rest as the training dataset.

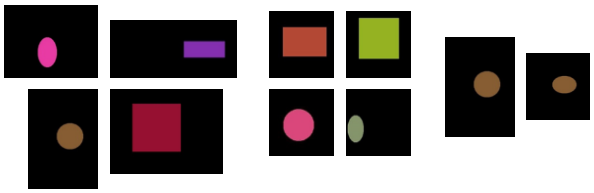

Figure 2. Example image of an unfixed-size geometry image dataset (left); Eample image of a fixed-size geometry image dataset (middle); Comparison between geometry image and warping image with geometric distortion (right).

It is worth noting that some images originally belong to a specific class. But after warping, the class corresponding to the image content has changed, as shown in Figure 2(right). This implies that the geometry image dataset is sensitive to geometric distortion.

III. VARIABLE STEP POOLING

A. *Variable Step Operators*

After dividing the feature map into a fixed number of partitions in the horizontal and vertical directions, SPP performs maximum pooling on each partition, and then splicing the pooled results into vectors, thereby receiving various size input and generating fixed-size output. However, the SPP partitioning method destroys the association among partitions. Therefore, SPP needs to divide the feature map into several different scales, and then all the outputs are spliced to a final output, thus brings more computational overhead. In the improvement of the model, we need to avoid unnecessary computational overhead as much as possible. Compared with SPP, VSP can simulate normal pooling better using only once partition, which means that the calculation amount is greatly reduced under almost the same effect, which is the main reason why we choose VSP in subsequent solution.

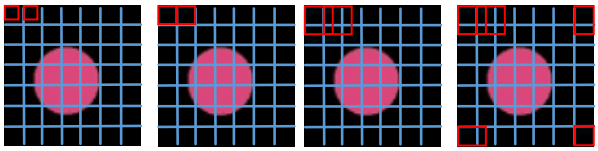

Figure 3. From left to right: the case when kernel size smaller than partition size; the case when kernel size is equal to partition size; the case when kernel size is larger than partition size; and the situation of underside/right border when kernel size is larger than partition size.

VSP uses the same partitioning method as SPP. It is required to set up the number of partitions in the horizontal and vertical directions, and also needs to set up the size of the kernel. When the kernel size is larger than the partition submap size, the effect is the same as the normal pooling with the common settings. When the kernel size is equal to the partition submap size, it is the same as the SPP and the normal pooling with some other settings. Within each partition, in addition to carry out the maximum pooling, the VSP can perform average or minimum pooling. When the pooling operation in each partition is replaced by convolution, the VSC is obtained. VSC can simulate normal convolution by setting, and also realizes accepting variable size input and generating fixed-size output. VSP and VSC will splice the processing results of the partition

submaps in situ, and obtain feature maps that can continue to use convolution or pooling to learn, which makes them suitable for more flexible application scenarios than SPP.

In theory, the normal pooling and normal convolution need to set the kernel size and the sliding step size. Meanwhile, the sliding times is related to the input size. VSP and VSC also need to set the kernel size, the difference is that the sliding times needs to be set, but the sliding step size is changed with the input size. The output size is the same as the horizontal and vertical sliding step size set, regardless of the input size. The sliding step size of VSP and VSC is variable, so we call them variable step operators.

B. Adding Coordinate Channels

We are inspired by Uber's experience in map processing [23]. Before using VSP or VSC to process feature maps, we can add coordinate channels to the feature map. The X-coordinates means the abscissa of the elements in the feature map, and the Y-coordinates means the ordinates of the elements in the feature map. The bottom left corner of the feature map is the origin of the coordinate system. The X/Y-coordinate channels are concatenated to the rear of existing channels, as shown in Figure 4.

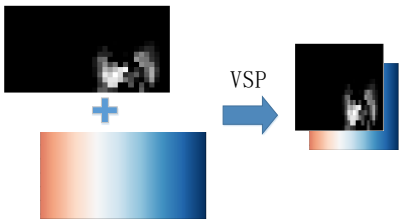

Figure 4. Adding an X-coordinate channel.

It is conceivable that this kind of adding coordinate method is very effective in the positioning task. However, it not only has no advantage in the classification task to add coordinate, but also makes some interference in the learning of normal information. Therefore, whether to add coordinate channels should be dependent on the task type.

IV. An Improvement for AlexNet to Accept Various Size Inputs

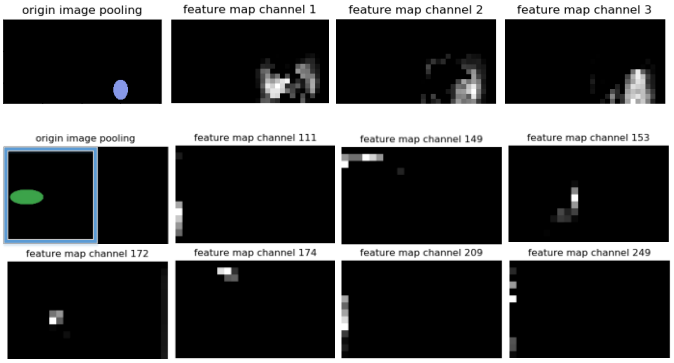

Figure 5. The input after sampling and the feature maps of last convolution in AlexNet before training (top); The prediction coordinate box on input after sampling and the feature maps of last convolution in V-AlexNet after training (bottom).

AlexNet [3] is a excellent classic model with a fully connected layer. We visualize the output of last convolution layer to get Figure 5(top). It shows that the effective pixels of all channels in output concentrated in an approximate region that is related to the original input. We define the last convolution output in the original model as M. This inspires us that we can extract a feature map block with the same shape containing as much effective information as possible from the feature maps obtaining with various sizes. Then VSP can be used to transform each feature map block to the same size without geometric distortion. Finally the new feature map is inputted to the fully connected layers of AlexNet for prediction of the probability distribution vector. We used this VSP-based solution to adapt AlexNet to accept various size inputs while retaining fully-connected layer, as is shown in Figure 6.

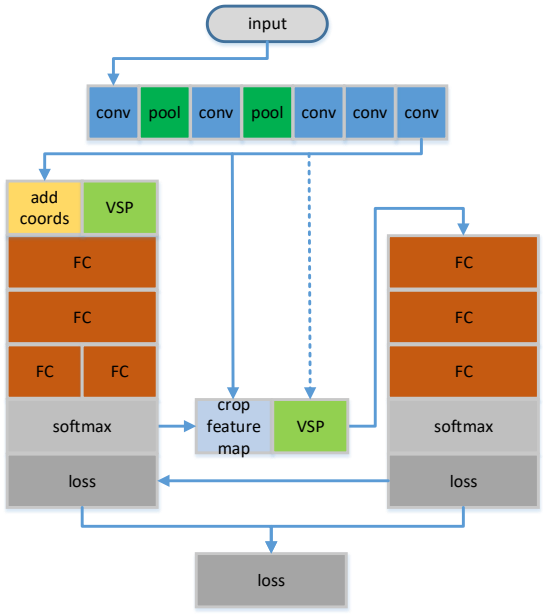

Figure 6. V-AlexNet structure.

In V-AlexNet, the coordinate prediction network is a fully-connected network similar to AlexNet's fully-connected network. The input length of the first layer is set to the length of the VSP layer output straightened, and the last layer includes two parallel fully-connected layer with same inputs, and the output lengths are respectively set to the maximum width W_{MAX} and the maximum height H_{MAX} of M, which has many possible sizes. These two parallel layers are respectively used to predict X-coordinate and Y-coordinate of the center point in the effective information area in M. The outputs of two last fully-connected layer are processed by the softmax function to obtain the probability distribution vectors of coordinates.

The solution can be summarized into the following steps:

Step 1: Calculate the maximum size W_{MAX} and H_{MAX} of M in all iterations and the size W and H of M in the current iteration. When $W \neq H$, then goto Step 2; when $W = H$, then goto Step 6;

Step 2: Add coordinate channels to M and perform VSP;

Step 3: Calculate the probability distribution vector of the coordinates according to the coordinate prediction network designed by W_{MAX} , H_{MAX} and VSP output sizes;

Step 4: According to the probability distribution vector of the X/Y-coordinates, the X-coordinates of top K probability and Y-coordinates of top K probability are obtained. When $W > H$, the Y-coordinate with the highest probability and each one of the obtained K X-coordinates form the K pair coordinates; when $W < H$, the X-coordinate with the highest probability and each one of the K Y-coordinates form the K pair coordinates;

Step 5: For each group of valid information area center point coordinates X_C and Y_C , when $W > H$, we use the vertical line corresponding to X_C as the cropping center line, and crop $H/2$ width range on both sides of the vertical line to get a size $H \times H$ feature map block. When $W < H$, we use the horizontal line corresponding to Y_C as the clipping center line, and crop $W/2$ width range on both sides of the horizontal line to get a size $W \times W$ feature map block . If the boundary exceeds during cropping, the map boundary is cropping boundary, as shown in Figure 7. Finally, K feature map blocks are obtained, and their center coordinates is recorded;

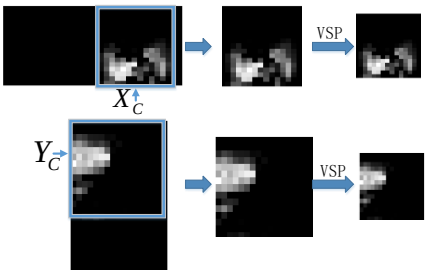

Figure 7. Extracting feature map effective information area.

Step 6: When $W = H$, directly perform VSP processing on M to obtain a new feature map, as shown by the dashed line in Figure 6; When $W \neq H$, the VSP processing of the same set is performed on each of the obtained feature map blocks, and K new feature maps are obtained, and corresponding relationship between the new feature map and the coordinates is recorded;

Step 7: When $W = H$, for the new feature map obtained, after the finish of fully-connected layer in the original model, we calculate the original model loss function to get the loss value $loss$, then goto Step 9; when $W \neq H$, for each new feature map obtained, after the finish of fully-connected layer in the original model, we calculate the original model loss function to get K losses and the minimum loss value $loss_{MIN}$. According to the corresponding relationship between the new feature map and the coordinates, we get the coordinates X_{MIN} and Y_{MIN} corresponding to the $loss_{MIN}$, then goto Step 8;

Step 8: In coordinate prediction network, the loss function uses X_{MIN} and Y_{MIN} as labels to calculate the sum of two cross entropy losses according to the probability distribution vector of two coordinates. The formula is:

$$loss_{XY} = -(\sum_{i=1}^{W_{MAX}} q_X(X_i)\log(p_X(X_i)) + \sum_{j=1}^{H_{MAX}} q_Y(Y_j)\log(p_Y(Y_j)))'$$

where $p_X(X_i)$ is the probability of X_i from the prediction probability, and $q_X(X_i)$ is the true probability value;

Step 9: When $W = H$, use $loss$ as the final loss; when $W \neq H$, the sum of $loss_{MIN}$ and $loss_{XY}$ is the final loss:

$$loss = loss_{MIN} + loss_{XY}.$$

In prediction, the probability prediction vectors corresponding to the X/Y-coordinates with top 1 probability are the predicted results.After the steps above, AlexNet is adapted to V-AlexNet, which can accept various size inputs. The whole improvement is completed in the model design stage. Other requirements for model training and prediction are constant, including consistent input size constraints within an iterative batch. From the network structure, V-AlexNet adds coordinate prediction and extraction of new feature maps, which brings time consuming, while the computing of K loss in the end of network can be parallel which does not bring significant consumption. In the prediction, the time consuming of single coordinate prediction network can be ignored. The solution can be easily transplanted in other models.

V. EXPERIMENTS ON GEOMETRY IMAGE DATASET

In this chapter and subsequent chapters, we do experiments on a GPU cluster with 28 nodes, each of which included an Intel Xeon CPU and an NVIDIA Tesla K80 GPU. We code in the software environment of TensorFlow 1.3 and python2.7.

A. Task Verification Experiment

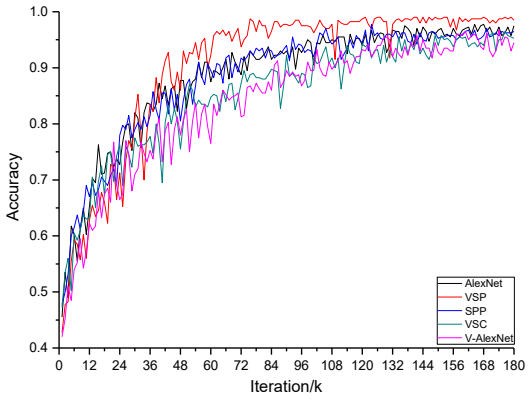

Figure 8. Task verification experiment on fixed-size geometry image dataset.

In the task verification experiment in this section, each training result is a single training result with no special tuning. It is only necessary to verify whether the geometry image can be effectively classified by CNN. The experiment settings are consistent to set learnning rate 1e-5 and set the minibatch 1. And we train the data 50 epoch using Adam optimizer.

On the fixed-size geometry image dataset, we experimented with AlexNet, AlexNet with SPP, AlexNet with VSP, AlexNet

with VSC, and V-AlexNet (K=1), respectively. It can be found that all the best accuracy are over 0.96, and they all have a good convergence effect. This shows that the geometry image dataset can be effectively classified by CNN, also shows that each method can be used for fixed-size image classification.

B. Experiments on Unfixed-Size Geometry Datasets

We compared the performance of V-AlexNet on unfixed-size geometry image dataset in different experimental settings. Each of results is the best one of the four repetitive trainings. The results are shown in TABLE I.

TABLE I. RESULTS OF V-ALEXNET WITH DIFFERENT SETTINGS ON UNFIXED-SIZE GEOMETRY IMAGE DATASET

K	Accuarcy with Adding Coordinate Channel	Accuarcy without Adding Coordinate Channel
1	0.56	0.5275
2	0.815	0.7325
3	0.895	0.885
4	0.91	0.8775
5	0.7925	0.6925

From TABLE I, we can see that V-AlexNet performs best when K=3 or 4, and adding coordinate channels in coordinate prediction can improve convergence stability and achieve better accuracy.

Then we do the experiments of AlexNet, AlexNet with SPP, AlexNet with VSP and AlexNet with VSC to compared with the experiments of V-AlexNet (K=4) . For AlexNet, the images are resized to 256×256. The situation after training 100 epoch is shown in Figure 9.

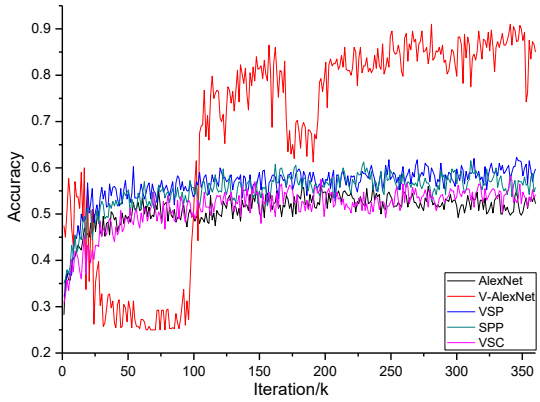

Figure 9. Results on unfixed-size geometry image dataset.

From the figure, AlexNet which uses warping, AlexNet with SPP, AlexNet with VSP, and AlexNet with VSC are extremely poor due to geometric distortion problems, while V-AlexNet (K=4) performs excellently. V-AlexNet can achieve an accuracy of 0.91. In terms of efficiency, due to more calculations, SPPNet and V-AlexNet training takes longer time than AlexNet.

We visualize the output of last convolution layer in V-AlexNet to get Figure 5(bottom). The effective pixels of channels whose pixels are not all zero are concentrated in the prediction coordinate box range, which means V-AlexNet can indeed predict by extracting effective information regions.

C. Verification Experiment on Other Model

In order to verify that our method is still valid on other models, we apply our method on an 11-layer VGGNet (A-LRN) [24]. We perform a comparison experiment of VGGNet with warping, VGGNet with SPP and VGGNet with our method on the unfixed-size geometry image dataset. But VGGNet with warping, VGGNet with SPP are difficult to learn due to geometric distortion. We perform multiple tests under various parameter settings, and their accuracy always dropped rapidly to the random classification accuracy value after weak oscillation at the beginning. On the contrary, VGGNet with our method can get a good result with 0.925 accuracy after only one training. The comparison results presented in Figure 10 verify that our solution is robust to models.

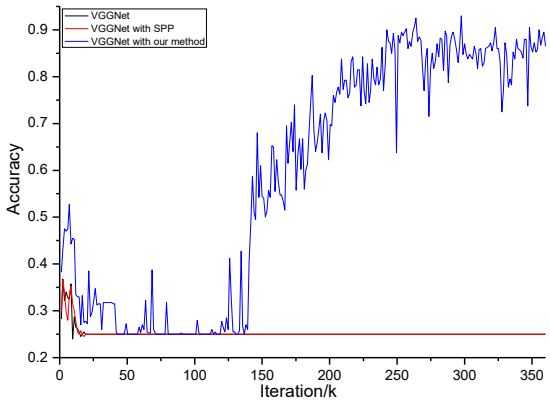

Figure 10. Experiment based on VGGNet (A-LRN) model.

VI. EXPERIMENTS ON STANDARD DATASETS

A. Classification Experiments on VOC 2007

The classification task in PASCAL VOC 2007 [20] contains a total of 9963 images in 20 classes, of which 5011 images are used for training, the rest are used for testing, and the mean average precision (mAP) is used to evaluate performance. As a standard dataset, VOC 2007 is the benchmark for measuring image classification and recognition. VOC 2007 is almost insensitive to geometric distortion, in which the size of the images is various, with the size about 500×375 or 375×500.In training, the images are resized so that the shorter side is 224.

We conduct further experiments based on the existing work of SPPNet. SPPNet is based on ZFNet [4] which is a variant of AlexNet. It is easy to transplant the solution on AlexNet to ZFNet. Our experiments on VOC 2007 use the experimental setup consistent with the previous work [4][14]. We get the results in TABLE II.

From the table we can see that all the methods have the normal performance for a general dataset that is not sensitive to geometric distortion. VSC can get similar results with ZFNet. Although it is difficult to improve, it is an option in the general dataset. Thanks to the sufficient image information, VSP can achieve better results. It is difficult to achieve the same effect as SPPNet does because there is no multi-level pooling, but there is almost no extra cost for VSP promotion, which is

important for real time tasks. ZFNet with our solution maintains the accuracy with original ZFNet. It is a normal result for dataset with no geometric distortion sensitivity. In short, our solution has normal performance on general natural image datasets that are not fixed in size.

TABLE II. RESULTS ON VOC 2007

Model	mAP
ZFNet	75.90
SPPNet(multi-level pooling)	78.39
ZFNet with VSC	75.04
ZFNet with VSP	76.37
ZFNet with Our Method	75.98

B. Classification Experiments on ImageNet

We use V-AlexNet (K=1) to train the ILSVRC-2010 and ILSVRC-2012 [25] , comparing with AlexNet's past experiments [3]. The results are shown in TABLE III. Our training setup follows the practice of previous work.

TABLE III. RESULTS ON IMAGENET

Dataset	AlexNet	V-AlexNet
ILSVRC-2010	top-1 37.5%, top-5 17.0%	top-1 37.4%, top-5 17.0%
ILSVRC-2012	top-5 15.3%	top-5 15.2%

From the above experiments, we can see that our solution maintains the accuracy of the original model on a fixed-size data set. However, in theory, our method is almost the same as the original model when dealing with fixed-size dataset. So we do not recommend using our solution for fixed-size dataset.

VII. CONCLUSION

In this paper, we build a geometry image dataset for the geometric distortion problem caused by the current data processing of warping. Then we propose two improvements of SPP called VSP and VSC that can be applicable to more scenarios. We also verify that the partition processing operator such as SPP still has geometric distortion problem and propose a VSP-based solution that allows CNNs with fully connection to accept various size inputs and learns effectively with no geometric distortion problems. The solution performs very well on geometry image dataset with geometric distortion sensitivity. Our solution can be easily transplanted to other models and other image processing tasks. In this article, we try in the single target image classification task. In the future, we will try to experiment, apply and optimize in other tasks.

ACKNOWLEDGMENT

This work was supported by National Key R&D Program of China 2018YFB0203904; National Natural Science Foundation of China U1611261, 61433019, and 61872392; and the Program for Guangdong Introducing Innovative and Enterpreneurial Teams under Grant NO. 2016ZT06D211.

REFERENCES

[1] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, “Backpropagation applied to handwritten zip code recognition,” Neural computation, 1989.

[2] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei- Fei, “Imagenet: A large-scale hierarchical image database,” in CVPR, 2009.

[3] A. Krizhevsky, I. Sutskever, and G. Hinton, “Imagenet classification with deep convolutional neural networks,” in NIPS, 2012.

[4] M. D. Zeiler and R. Fergus, “Visualizing and understanding convolutional neural networks,” arXiv:1311.2901, 2013.

[5] Leroux, Sam, et al. "IamNN: Iterative and Adaptive Mobile Neural Network for Efficient Image Classification." (2018).

[6] Zhang, Zhi, et al. "Progressive Neural Networks for Image Classification." (2018).

[7] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmen- tation,” in CVPR, 2014.

[8] Inoue, Naoto, et al. "Cross-Domain Weakly-Supervised Object Detection through Progressive Domain Adaptation." (2018).

[9] Chen, Yuhua, et al. "Domain Adaptive Faster R-CNN for Object Detection in the Wild." (2018).

[10] Zhang, Yiheng, et al. "Fully Convolutional Adaptation Networks for Semantic Segmentation." (2018).

[11] Yang, Ren, et al. "Multi-Frame Quality Enhancement for Compressed Video." (2018).

[12] Li, Lerenhan, et al. "Learning a Discriminative Prior for Blind Image Deblurring." (2018).

[13] Fan, Lijie, et al. "End-to-End Learning of Motion Representation for Video Understanding." (2018).

[14] He K, Zhang X, Ren S, et al. Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition.[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2015, 37(9):1904-1916.

[15] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell, “Decaf: A deep convolutional activation feature for generic visual recognition,” arXiv:1310.1531, 2013.

[16] Lin M, Chen Q, Yan S, “Network In Network”, Computer Science, 2013.

[17] K. Grauman and T. Darrell, “The pyramid match kernel: Discriminative classification with sets of image features,” in ICCV, 2005.

[18] S. Lazebnik, C. Schmid, and J. Ponce, “Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories,” in CVPR, 2006.

[19] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein et al., “Imagenet large scale visual recognition challenge,” arXiv:1409.0575, 2014.

[20] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, “The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results,” 2007.

[21] Xu, Mengjia, et al. "A deep convolutional neural network for classification of red blood cells in sickle cell anemia." *PLOS Computational Biology* 13.10 (2017).

[22] PLOS. "New machine learning system can automatically identify shapes of red blood cells: Deep learning approach could aid in sickle cell disease monitoring." ScienceDaily. ScienceDaily, 19 October 2017.

[23] Liu, Rosanne, et al. "An intriguing failing of convolutional neural networks and the CoordConv solution." neural information processing systems (2018).

[24] Simonyan, Karen, and Andrew Zisserman. "Very Deep Convolutional Networks for Large-Scale Image Recognition." international conference on learning representations (2015).

[25] A. Berg, J. Deng, and L. Fei-Fei. Large scale visual recognition challenge 2010. www.imagenet.org/challenges. 2010.