A novel CNN model for fine-grained classification with large spatial variants

Junpeng Wang¹², Yanfeng Lu¹*

¹ The State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing, 100190, China
² School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing, 100049, China
*yanfeng.lv@ia.ac.cn

Abstract. Convolutional Neural Networks (CNN) have achieved great performance in many visual tasks. However, CNN models are sensitive to samples with large spatial variants, especially severe in fine-grained classification task. In this paper, we propose a novel CNN model called ST-BCNN to solve these problems. ST-BCNN contains two functional CNN modules: Spatial Transform Network (STN) and Bilinear CNN(BCNN). Firstly, STN module is used to select key region in input samples and get it spatially modified. Since the adoption of STN will cause an information loss phenomenon called boundary loss, we design a brand-new IOU loss method to solve it. We make a theoretical analysis of the IOU loss method. Secondly, to discover discriminative features for fine-grained classification task, BCNN module is applied. BCNN interacts CNN features from different channels to produce more discriminative bilinear features than fully connected features of CNN. ST-BCNN works by reducing irrelevant spatial states and producing fine-grained features. We evaluate our model on 3 public fine-grained classification datasets with large spatial variants: CUB200-2011, Fish100 and UAV43. Experiments show that the IOU loss method can reduce boundary loss and make STN module output spatial transformed image appropriately. Our proposed ST-BCNN model outperforms other advanced CNN models on all three datasets.

1. Introduction
Over recent years, the study of computer vision tasks has been pushed forward by the development of deep learning algorithms. Convolutional Neural Networks (CNN) is a primary deep learning algorithm [1]. With the connection of multiple learnable convolutional and fully connected layers, it can extract more representative features than traditional human-designed features. Despite these progresses, there’re still some drawbacks in CNN. It is sensitive to samples with large spatial variants, especially severe in fine-grained classification task.

1.1. Task with large spatial variants
CNN has weak abilities dealing with spatial variants. Spatial pooling operations can achieve spatial translation tolerance within its pooling kernel. However, if the spatial variants are relatively large, it doesn’t work. The pooling operations will lose details of the image, which may be useful in fine-grained tasks. What's more, CNN can't deal with samples with large spatial rotation very well.

Some studies aim to solve this problem. Spatial transformer network (STN) is an initiative work [2]. This network outputs spatial transformed features based on input features. It can work as a subnet
module in any large network. It allows end-to-end training. However, due to end-to-end training strategy, it will sometimes not converge and have boundary loss.

A modified architecture is Inverse Compositional spatial transformer network (IC-STN) [3]. Different from traditional STN, it uses cascaded STNs to predict the transformation. The next STN takes the output of previous STN as input. The final transformation is multiplied by all relevant STN transformations. This network improves the classification performance but still suffers from boundary loss.

1.2. Fine-grained classification task

For classification tasks, many researches of CNN focus on coarse-grained classification. But fine-grained classification is more challenging. Fine-grained classification means classifying samples from the same class into different subclasses. The samples may be very similar and may differ in just subtle parts. For samples in the same subclass, they may have very different appearance due to spatial variants. In all, fine-grained samples have small inter-class invariance and large intra-class variance.

For fine-grained classification task, there have been some studies. The studies can be divided into two types of methods: strongly-supervised method and weakly-supervised method. Strongly-supervised method needs manual object part annotations. Since samples differ in just subtle parts, comparing parts between samples rather than the entirety is realistic [4]. Major methods include part-based R-CNN [5], Mask-CNN [6] and Pose Normalized CNN [7]. Although the performance of them is better than weakly-supervised method, it’s time-consuming to annotate object parts.

Without extra labels, weakly-supervised method can be trained just on class label. There exist two main types: attention-based type and bilinear pooling type. With the mechanism of attention, the weakly-supervised method can automatically detect important object parts. Recurrent Attention CNN [8] uses an Attention Proposal Network (APN) module to zoom in key parts recurrently. Multi-Attention CNN (MA-CNN) [9] divides the last convolutional layers into groups, each group means a part attention. Other important works include MAMC [10], RAM [11] and RAN [12]. Bilinear pooling [13-14] is a method to interact convolutional features on different channels. It digs into the relationship between different convolutional features since features from different convolutional channels are very rich [15-16]. It also has some variants: compact bilinear pooling, hierarchy bilinear pooling etc. However, these weakly-supervised methods don’t focus on the problem of spatial variants. When the spatial variants increase, their performance will be worse.

In this paper, we aim to solve the above problems. By introducing a novel IOU loss, we solve the problem of boundary loss in STN, making the performance of STN more stable. By combining STN and BCNN, we design a novel network ST-BCNN for fine-grained classification with large spatial variants. It works by comparing samples under similar spatial states and digging into fine-grained features. It outperforms other advanced CNN models on three datasets.

2. Related works

2.1. Spatial transformer network (STN)
2.1.1. Structure of STN. STN outputs spatial transformed features based on input features (see Figure 1) [2]. Assuming the coordinates of an input point \( p_i \) is \((x_i^s, y_i^s)\) and its transformed point \( p_i \) is \((x_i^t, y_i^t)\). The coordinates of \( p_i^s \) are normalized such that \(-1 \leq x_i^s, y_i^s \leq 1\). The relationship is as (1) and \( A_\theta \) is the transformation matrix. Parameters \( a, b, c \) and \( d \) are used to calculate scale, rotation, flip and shear transformation while \( t_1 \) and \( t_2 \) are used to calculate translation.

\[
\begin{bmatrix}
  x_i^t \\
  y_i^t 
\end{bmatrix} = A_\theta \begin{bmatrix}
  x_i^s \\
  y_i^s 
\end{bmatrix} = \begin{bmatrix}
  a & b & t_1 \\
  c & d & t_2 
\end{bmatrix} \begin{bmatrix}
  x_i^s \\
  y_i^s \\
  1 
\end{bmatrix} \tag{1}
\]

STN contains a localization network. This localization network takes the input feature and outputs a 6-dimensional vector, the vectors is reshaped into transformation matrix \( A_\theta \). The localization network can be in any structure as long as the output is a 6-dimensional vector. A common solution is a CNN with several convolutional layers and fully-connected layers.

Suppose that input feature map \( I \in \mathbb{R}^{H \times W \times C} \) is with height \( H \), width \( W \) and \( C \) channels. Output feature map \( O \in \mathbb{R}^{H' \times W' \times C} \) have the same channels \( C \) but different height \( H' \) and width \( W' \). A sampling kernel is used to get the value at a particular pixel in the output \( O \) [2]. The form of sampling kernel is (2). Since the coordinates of output feature pixels must be integers. It is approximated by its nearby transformed points.

\[
O_i^c = \sum_n \sum_m I_{nm} \max(0, 1 - |x_i^t - m|) \max(0, 1 - |y_i^t - n|) \quad \forall i \in [1, \ldots, H'] \quad \forall c \in [1, \ldots, C] \tag{2}
\]

2.1.2. Boundary loss. Since STN takes an end-to-end framework. Thus the training process is usually unstable. The outputs of STN will suffer a problem called boundary loss (see Figure 2). The output image samples only from the cropped image. Pixel information outside the crop region is discarded, causing the boundary of the output image to be empty. CNN can’t deal with the empty boundary very well. Boundary loss sometimes makes the performance of a CNN with STN module even worse than original CNN.

![Figure 2. Boundary loss](image)

2.2. Bilinear CNN

Bilinear CNN(BCNN) can model local pairwise feature interactions (see Figure 3) [13]. It interacts features from different channels. A CNN is a feature mapping \( f: I \times L \rightarrow \mathbb{R}^{c \times D} \). It takes an image \( I \) and localization \( L \), and outputs a feature map with \( c \) channels and dimensions \( D \) in each channel. The location
\[ L \text{ is referred generally which can include position and scale [13]. The bilinear feature combination at each position } l \text{ is the matrix outer product as (3):} \]

\[ (l, l, f_A, f_B) = f_A(l, l)^T f_B(l, l) \]  

(3)

The bilinear image descriptor \( \Phi(I) \) is gotten by a pooling method of bilinear features at different locations. A common pooling method is sum pooling over all locations.

\[ \Phi(I) = \sum f_A(l, I)^T f_B(l, I) \]  

(4)

The descriptor is followed by a signed square root step \( y = \text{sign}(\Phi(I)) \sqrt{|\Phi(I)|} \) and a l2-norm normalization. The bilinear feature combination and sum pooling method are all differentiable. The bilinear CNN model can be optimized with end-to-end training.

Structures of BCNN model can be divided into fully shared, partially shared and no sharing [14]. In fully shared model (Figure 4(a)), only one CNN model is used, \( f_A(l, I) \) and \( f_B(l, I) \) are the same feature mapping. In no sharing model (Figure 4(b)), two different CNNs are used to extract features. In partially shared model (Figure 4(c)), a part of CNN model is shared.

![Figure 4. Structures of BCNN](image)

3. ST-BCNN

3.1. IOU loss

Intersection over Union (IOU) reviews intersecting area formed by original (Figure 5(a)) and transformed images (Figure 5(b)). More information will remain after transformation with higher IOU value. The green part in Figure 5(c) is the IOU area. There will be less information loss with larger IOU area. The maximum IOU value is 1.0 (Figure 5(d)). In this case, there will be no boundary loss.

![Figure 5. IOU area](image)

However, the shape of transformed image is usually irregular quadrilateral, making the IOU value hard to calculate. We introduce an approximate calculation method to calculate IOU value. If more points on the boundary of transformed image is outside the area of original image, there will usually be higher IOU value. In this case, we inspect positions of these boundary points.

Since there’re infinite points on the boundary. We select eight key points based on the flowing principle (see Figure 6). In figure 6, purple point A is a point on original image and red point A’ is its transformed point on transformed image. 4 corner points A (1, 1), B (-1, 1), C (-1, -1) and D (1, -1) are important since they decide the shape of transformed image. Another 4 points E, F, G and H are uniform sampled from four edges.

We suppose the coordinates of an input point \( p^i \) and its transformed point \( p^f \) are \( (x^i, y^i) \) and \( (x^f, y^f) \) respectively. If \( p^f \) is outside the area of original image, at least one of \( x^f \) and \( y^f \) will be greater than 1. However, if the value is infinitely great, the transformed image will be amplified too much. Considering these factors, an IOU loss is established as (5):
\[ L_{IOU} = \sum_{i=1}^{8} \text{ReLU}(1 - |p_i|) \]

Figure 6. Eight key points

Inspired by ReLU loss \cite{21}, We establish IOU loss. If the infinite norm is less than 1, The loss will be greater than 0. But the infinite norm is nondifferentiable and can’t be optimized by back propagation. We modified the loss into (6).

\[ L_{IOU} = \sum_{i=1}^{8} \left[ \text{ReLU}(1 - |x_i|) + \text{ReLU}(1 - |y_i|) \right] \]

A single point form of IOU loss can be extended with original point coordinates (7). The derivation form of (7) is (8) and (9). The derivation about parameter \( b \), \( c \), \( d \) is in the same form of parameter \( a \), while that about parameter \( t_2 \) is in the same form of parameter \( t_1 \). Since IOU loss is differentiable about all six parameters, it can be adopted to train CNN with back propagation.

\[ L_{IOU} = \text{ReLU}(1 - |x_i|) + \text{ReLU}(1 - |y_i|) \]

\[ \frac{\partial L_{IOU}}{\partial a} = \begin{cases} -x_i, & 0 \leq ax_i^s + by_i^s + t_1 \leq 1 \\ x_i, & -1 \leq ax_i^s + by_i^s + t_1 \leq 0 \\ 0, & \text{else} \end{cases} \]

\[ \frac{\partial L_{IOU}}{\partial t_1} = \begin{cases} -1, & 0 \leq ax_i^s + by_i^s + t_1 \leq 1 \\ 1, & -1 \leq ax_i^s + by_i^s + t_1 \leq 0 \\ 0, & \text{else} \end{cases} \]

3.2. Details of ST-BCNN

Spatial variants make CNN produce many redundant features, which expends feature space. If we can reduce spatial variants, thus the irrelevant subspace is reduced. So, we want to transform the sample before it’s fed into CNN. Features from different channels of CNN are relevant in some dimensions, bilinear pooling can help to find the relationship between them. In fact, bilinear pooling can be viewed
as a coupled feature transformation method to produce more discriminative features for fine-grained task.

Inspired by the ideas above, we adopt two functional CNN modules: spatial transformer network and Bilinear CNN together to get a novel network called ST-BCNN. Firstly, the input image is transformed by a STN module, thus the image is spatially modified and key part is focused. Then, the transformed image is processed by a CNN module. Finally, the output convolutional features of CNN module are processed by bilinear pooling. After a softmax layer, a class is predicted. (see Figure 7).

It is worth mentioning that in ST-BCNN, we adopt fully-sharing BCNN, only one CNN model is used to extract features. Compared with no-sharing BCNN, it costs about half operation time. In bilinear pooling operation, only the last convolutional layer of CNN is used. Features from different channels are interacted by an outer-product operation. If there’re c channels, the bilinear feature has \( c^2 \) dimensions.

In training process, the total loss is composed of two losses (10). The first part is cross entropy loss, it’s used to measure classification accuracy. The second part is IOU loss, which is used to reduce information loss in the spatial transform process. A parameter \( \alpha \) is used to balance these two losses.

\[
\min(Loss) = \min\left(L_{cross entropy} + \alpha L_{IOU}\right)
\]

To train the model, we use a two-stage training method. Firstly, STN module is combined with CNN module. Only these two modules are trained. Secondly, the fully connected layers of CNN are replaced with Bilinear pooling layers, we finetune parameters of the whole model.

3.3. Baseline: Inception Resnet V2

We select Inception Resnet V2 [17] as our baseline CNN module. It combines both Resnet connection and inception module. On one hand, Resnet connection deepens CNN depth. On the other hand, inception module uses multi-scale receptive fields to produce multi-scale features. It performs better on ImageNet dataset than other famous CNN structure like Resnet [19] and Inception-v4 [18].

4. Experiments

Our STN architecture is relatively simple. It takes original image as input. It consists 2 convolutional layers, 3 max-pooling layers and 2 fully connected layers. It outputs a 6-dimensional vector. These 6 elements are parameters of transformation matrix.

| Datasets        | Classes | Training | Testing |
|-----------------|---------|----------|---------|
| CUB200-2011     | 200     | 9431     | 2357    |
| Fish100         | 100     | 5087     | 1271    |
| UAV43           | 43      | 2477     | 619     |

We test our ST-BCNN on 3 datasets: CUB200-2011 bird datasets [20], Fish100 datasets [23], and an Unmanned Aerial Vehicle (UAV) dataset established by ourselves. They are all fine-grained classification datasets. Besides, samples of them all have large spatial variants. We split 80% images of each dataset into training sets, while the rest is for testing.

4.1. Parameter selection

The parameter \( \alpha \) in loss will affect STN accuracy. We conduct an experiment to find the optimal \( \alpha \) value on CUB200-2011 dataset.

The experiment result is in Table 2. Performance of our baseline Inception Resnet V2 is 78.17%. If IOU loss is not applied, adding a STN module even worsens the accuracy, which is 66.17%. By applying IOU loss, the performance is better than the baseline. When the value of \( \alpha \) is 5.0, the best performance 83.42% is achieved.

Figure 8 denotes the value of IOU loss with respect to steps. Log value of IOU loss is used in Figure 8 for plotting consideration. If \( \alpha = 0.1 \), the loss can hardly converge to zero. Inversely, if \( \alpha \) is no less than 1.0, the loss can converge to zero. The convergence process will be faster if the value of \( \alpha \) is greater.
However, if $\alpha$ is too large, less classification information is used during training. In all, 5.0 is a balanced value for $\alpha$.

Table 2. Results of parameter selection experiment

| Model                                | Accuracy   |
|--------------------------------------|------------|
| Inception Resnet V2                 | 78.17%     |
| Inception Resnet V2+STN              | 66.17%     |
| Inception Resnet V2+STN ($\alpha=0.1$) | 81.88%     |
| Inception Resnet V2+STN ($\alpha=1.0$) | 83.25%     |
| Inception Resnet V2+STN ($\alpha=5.0$) | **83.42%** |
| Inception Resnet V2+STN ($\alpha=8.0$) | 83.38%     |
| Inception Resnet V2+STN ($\alpha=10.0$) | 82.20%     |

Table 3. Results of classification experiment

| Model                                | CUB200 | Fish100 | UAV43 |
|--------------------------------------|--------|---------|-------|
| Vgg-16                               | 72.30% | 79.93%  | 69.67%|
| Inception V3                         | 74.04% | 86.51%  | 73.58%|
| Inception Resnet V2                  | 78.17% | 87.77%  | 77.00%|
| BCNN                                 | 80.08% | 90.92%  | 80.09%|
| RAN                                  | 82.80% | 91.64%  | 80.89%|
| Inception Resnet V2+STN ($\alpha=5.0$) | 83.42% | 93.23%  | 85.14%|
| **ST-BCNN**                          | **84.21%** | **94.23%** | **86.08%** |

Classification result is in Table 3. We find that Inception Resnet V2 outperforms other classical CNN models in all 3 datasets. This proves Inception Resnet V2 is a suitable baseline model. All the specific fine-grained models perform better than classical CNN models. Among them, the baseline + STN using IOU loss ranks second. ST-BCNN model is the best one in all 3 datasets. It achieves around 1% accuracy higher than baseline + STN model.
Figure 9 denotes some examples of transformation made by STN module. By using IOU loss, the results avoid boundary loss: there is no empty area on transformed image. Transformed images focus on key area on input image like attention mechanism. In figure 9, objects are zoomed in by STN module. However, different from common attention method, STN module makes a translation modification (Figure 9(b)), making the classification performance better than common attention method.

![Figure 9. Some examples of STN transformation](image)

(a)Bird  (b)UAV  (c) Fish

(5. Conclusion)

In this paper, we propose a novel CNN called ST-BCNN to solve fine-grained classification with large spatial variants. ST-BCNN contains two functional CNN modules: Spatial Transform Network (STN) and Bilinear CNN(BCNN). Considering boundary loss in STN model, we design an IOU loss method. We make detailed analysis of IOU loss to prove it reasonable and differentiable. We use a parameter $\alpha$ to balance it with cross entropy loss. By comparing classification accuracy at different $\alpha$ value, we find when $\alpha = 5.0$, the model performs best. With IOU loss, STN module avoids boundary loss and makes appropriate spatial transformation. ST-BCNN model outperforms some state-of-the-art methods in different datasets. The accuracies on CUB200-2011, Fish100 and UAV43 datasets are 84.21%, 94.23% and 86.08% respectively. ST-BCNN has both advantages of STN and BCNN, making it better than STN. It concludes our model can solve fine-grained classification with large spatial variants very well.

Acknowledgements

This work is funded by National Natural Science Foundation of China (Grants 61603389, 91648205).

References

[1] KRIZHEVSKY, A., SUTSKEVER, I., and HINTON, G.E. (2012) Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems, pp. 1097-1105.

[2] JADERBERG, M., SIMONYAN, K., and ZISSERMAN, A. (2015) Spatial transformer networks.

In: Advances in neural information processing systems, pp. 2017-2025.

[3] LIN, C.-H. and LUCEY, S. (2017) Inverse compositional spatial transformer networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2568-2576.

[4] WEI, Z., YANG, W., HUANG, W., DAI, H., and LI, D. (2019) Person Re-Identification Based on Multi-Parts of Local Feature Network. IEEE Access 7: 132438-132447.

[5] ZHANG, N., DONAHUE, J., GIRSHICK, R., and DARRELL, T. (2014) Part-based R-CNNs for fine-grained category detection. In European conference on computer vision Springer: 834-849.

[6] WEI, X.-S., XIE, C.-W., and WU, J. (2016) Mask-cnn: Localizing parts and selecting descriptors for fine-grained image recognition. arXiv preprint arXiv:1605.06878.

[7] BRANSON, S., VAN HORN, G., BELONGIE, S., and PERONA, P. (2014) Bird species categorization using pose normalized deep convolutional nets. arXiv preprint arXiv:1406.2952.

[8] FU, J., ZHENG, H., and MEI, T. (2017) Look closer to see better: Recurrent attention convolutional neural network for fine-grained image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4438-4446.
[9] ZHENG, H., FU, J., MEI, T., and LUO, J. (2017) Learning multi-attention convolutional neural network for fine-grained image recognition. In: Proceedings of the IEEE international conference on computer vision. pp. 5209-5217.

[10] SUN, M., YUAN, Y., ZHOU, F., and DING, E. (2018) Multi-attention multi-class constraint for fine-grained image recognition. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 805-821.

[11] LI, Z., YANG, Y., LIU, X., ZHOU, F., WEN, S., and XU, W. (2017) Dynamic computational time for visual attention. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 1199-1209.

[12] WANG, F., JIANG, M., QIAN, C., YANG, S., LI, C., ZHANG, H., WANG, X., and TANG, X. (2017) Residual attention network for image classification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3156-3164.

[13] LIN, T.-Y., ROYCHOWDHURY, A., and MAJI, S. (2015) Bilinear cnn models for fine-grained visual recognition. In: Proceedings of the IEEE international conference on computer vision. pp. 1449-1457.

[14] LIN, T.-Y., ROYCHOWDHURY, A., and MAJI, S. (2017) Bilinear convolutional neural networks for fine-grained visual recognition. IEEE transactions on pattern analysis and machine intelligence, 40: 1309-1322.

[15] FANG, J., XU, X., LIU, H., and SUN, F. (2019) Local receptive field based extreme learning machine with three channels for histopathological image classification. International Journal of Machine Learning and Cybernetics, 10: 1437-1447.

[16] ZHENG, W., LIU, H., WANG, B., and SUN, F. (2019) Cross-modal learning for material perception using deep extreme learning machine. International Journal of Machine Learning and Cybernetics: 1-11.

[17] SZEGEDY, C., IOFFE, S., VANHOUCKE, V., and ALEMI, A.A. (2017) Inception-v4, inception-resnet and the impact of residual connections on learning. In: Thirty-First AAAI Conference on Artificial Intelligence.

[18] SZEGEDY, C., VANHOUCKE, V., IOFFE, S., SHLENS, J., and WOJNA, Z. (2016) Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2818-2826.

[19] HE, K., ZHANG, X., REN, S., and SUN, J. (2016) Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770-778.

[20] WAH, C., BRANSON, S., WELINDER, P., PERONA, P., and BELONGIE, S. (2011) The caltech-ucsd birds-200-2011 dataset.

[21] XU, B., WANG, N., CHEN, T., and LI, M. (2015) Empirical evaluation of rectified activations in convolutional network. arXiv preprint arXiv:1505.00853.

[22] SIMONYAN, K. and ZISSERMAN, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.

[23] JI, Z., ZHAO, K., ZHANG, S., and LI, M. (2019) Fine-grained Fish Image Classification Based on Bilinear Network with Spatial Transformation. Journal of Tianjin University, 52: 475-482.