Adaptive Target Region for Birds Classification

Feitong Tan
Department of Computing Science
Simon Fraser University
Burnaby, BC V5A1S6
feitongt@sfu.ca

Yejia Liu
Department of Computing Science
Simon Fraser University
Burnaby, BC V5A1S6
yejial@sfu.ca

Abstract

General Convolutional Neural Network(CNN) performs analysis on the entire image. However, inspired by Spatial Transformer Network[3], we realize that only using the target part of image rather than the whole image can improve accuracy of final classification. Thus, we present an end-to-end deep learning approach to utilize the information of target object region. We initialize a target region at first by training given bounding box from dataset. During the subsequent learning process, training adaptive target region is used as an auxiliary to the main task, object classification. Several experiments are carried out on CUB-200-2011 birds images dataset. The comparison results show that our model outperforms general Convolutional Neural Network.

1 Introduction

Convolutional Neural Network(CNN) has a long history in computer vision, with early examples showing successful results on using supervised back-propagation networks to perform digit recognition[1]. More recently, the convolutional network proposed by Krizhevsky et al.[2] has outperformed pervious models on large benchmark image datasets like ImageNet. Trained via back-propagation through convolutional layers along with on-line approximate model averaging (dropout)[2], in the field of classification, CNN has achieved many state-of-the-art results.

To further improve CNN, by alleviation its issue of spatial variance happened in convolutional layers, Spatial Transformer Network[3] is proposed. It can be dropped into CNN at any point and in any number to keep spatial invariance of input data. After adding Spatial Transformer Network into CNN, an image can be actively transformed by producing an appropriate transformation for each input sample. During transformation process, the entire feature map can be scaled, cropped, rotated, and deformed. This allows networks not only to select most relevant regions in an image, but also to transform those regions to a expected pose, which simplifies recognition in subsequent layers. However, we find out that the training of Spatial Transformer Network[3] can be unpredictable and hard to converge for small dataset or dataset with confusing-content images, due to its high freedom in learning process. To get better performance when applying Spatial Transformer, we initialize a target region on each image by training given bounding box in our dataset, to decrease the randomization of generating bounding box in training.

In our project, instead of directly applying spatial transformer to the initial image[3], we only apply it to an adaptive target region, which is estimated at first as an auxiliary task. Because intuitively, bounding only target region rather than the entire image can decrease noises of the whole image. In this whole end-to-end network, the object classification is the main task while target region prediction is as the auxiliary one. The network framework is illustrated in Figure 1.

The dataset which we apply our network to is Caltech-UCSD Birds-200-2011(http://www.vision.caltech.edu/visipedia/CUB-200-2011.html). It contains both ample im-

ages and rich annotations with bounding boxes, which is perfect to be used as our initial target region. Many methodologies and work[4][5] has been carried out on this dataset. Among all of them, deep learning version of Deformable Part Descriptors(DPD)[5] has achieved state-of-the-art results with 50.98% accuracy on the whole dataset. Compared to features calculated by SVM[4], deep learning works better in large datasets and reduces worries on features engineering parts. Thats also one of the reasons why we use deep learning methodology to classify birds images rather than SVM.

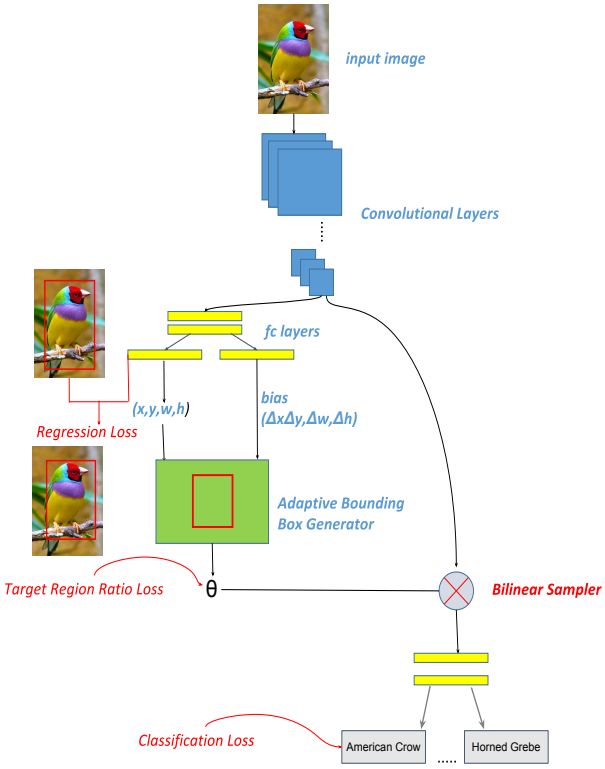


Figure 1: Overview of the Entire Network

To see whether our convolutional neural network adding Spatial Transformer contributes to classification scores, we compare its result with the results of using general pre-trained networks–AlexNet[2] and VGG[6], respectively. The comparison results show the efficacy of our end-to-end learning approach.

2 Approach

Our network architecture is illustrated in Figure 1. It consists of a convolutional network to extract features, an adaptive target region generator to predict target region, a target region based feature sampler and softmax classifier at last. They are refined as follows.

2.1 Convolutional Feature Extraction

A convolutional network aims to extract feature maps from a given image. These features are used for all subsequent tasks to enhance the computational efficiency. We apply both AlexNet and VGG

in our experiment to extract feature map.The images are resized to 227*227 as input for AlexNet while 224*224 for VGG.

2.2 Target Region Initialization

Intuitively, bounding only target region rather than analyzing the whole image can produce higher accuracy for classification. The initial target region is denoted as $b = [x, y, w, h]$, where (x, y) is the coordinate of upper left corner of initial target region, and w/h is the width/height of initial target region. In our experiment, b is trained based on given bounding box, which is easy to get from our dataset website(/urlhttp://www.vision.caltech.edu/visipedia/CUB-200-2011.html). As shown in Figure 2, after convolutional layers,we add three fully connected layers,with output dimensions 2048, 2048, 4, respectively. Each of them followed by a ReLU activation function and drop out layer(ratio 0.5), to predict target region. Note in the third fully connected layer, we use L2 distance loss to regress $b = [x, y, w, h]$ on given bounding box from dataset as ground truth.

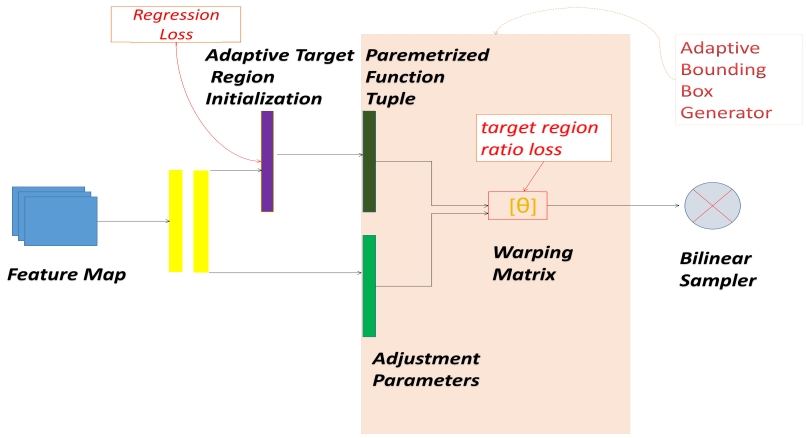


Figure 2: Details of Target Region Initialization and Generation

2.3 Adaptive Target Region Generation

The initial target region is then adaptively adjusted by free parameters $\Delta = [\Delta x, \Delta y, \Delta w, \Delta h]$ in network. The final target region is denoted as $[w(1+\Delta w), h(1+\Delta h), x(1+\Delta x), y(1+\Delta y)]$. To learn these free parameters, we add an additional fully connected layer to the previous layers to output four values.

In the learning process for free parameter, we noticed an problem that the final bounding box could be too distorted. To alleviate this issue,inspired by [7],we applied a target region ratio loss function as

$$L_r^t = \begin{cases} \frac{1}{2}\{[\alpha h_t(1+\Delta h)]^2 - [w_t(1+\Delta w]^2\}_+ & \text{if } h_t(1+\Delta h) > w_t(1+\Delta w) \\ \frac{1}{2}\{[\alpha w_t(1+\Delta w)]^2 - [h_t(1+\Delta h]^2\}_+ & \text{if } h_t(1+\Delta h) < w_t(1+\Delta w) \end{cases} \quad (1)$$

where α is a ratio threshold,which is 0.6 in our experiment.The loss is 0 when the value in bracket $\{\}_+$ is less than 0.

2.4 Bilinear Feature Sampling and Classification

We use a bilinear sampler introduced in[3] to wrap our target region via bilinear interpolation. It allows the gradient flow into the localization network to link the localization network and classifica-

tion layers. The warping uses a 2×3 affine transformation, as

$$\theta_t = \begin{bmatrix} w_t(1+\Delta w) & 0 & x_t + \Delta x \\ 0 & h_t(1+\Delta h) & y_t + \Delta y \end{bmatrix} \tag{2}$$

The transformation θ_t changes the coordinates in the target feature map U back to the coordinates (x,y) in the source feature map V. Specifically, we have

$$V_{(x,y)} = \sum_{m=1}^{W} \sum_{n=1}^{H} U_{(m,n)} max(0, 1-|x'-m|) max(0, 1-|y'-n|) \tag{3}$$

where $(x, y) \in R^2$ and $(m, n) \in R^2$ are coordinates in V and U, respectively. W/H are feature map dimensions. The $(x', y') \in R^2$ in U are warped coordinates which satisfy $[x'\ y']^T = \theta_t [x\ y\ 1]^T$. More details can be found in [3].

After bilinear feature sampling, there are two fully connected layers, each with a ReLU activation function and drop out layer(ratio (0.5)), to output dimensions 512 and 256. At the end of network, a softmax classifier is appended.

This whole network is end-to-end, using stochastic gradient descent. To train target region b, we record coordinates of upper left corner of bounding box and its width/height when feeding forward, then pass the errors back to proper channels during back-propagation.

3 Experiment

Our experiments are implemented through Caffe[8]. The strategy of tuning parameters in our network referred from ImageNet[9]. We set the initial base learning rate as 0.0006, weight decay as 0.0006. We train 160K iterations. For the adaptive target region training part, we use the same learning rate with the base learning rate. The weight of loss for target region generation(adjustments of free parameters) is 2.8. For target region ratio loss, the weight of loss is 0.05. In training phase, we set batch size as 64 for AlexNet and 10 for VGG.

3.1 Dataset

we evaluate our approach on CUB-200-2011 dataset, which is a common benchmark for object recognition and classification problems. It consists of 200 different bird categories, 11,788 images, each along with a bounding box, which is used for initial target region regression in our network. In contrast to the previous CUB-2010 dataset, the CUB-2011 dataset doubled the number of images per class to combat overfitting with larger dataset.

In our experiment, the whole dataset is divided into training set, validation set and testing set, by ratio 0.6,0.2, and 0.2, respectively. And our results is computed on the whole CUB-2011 dataset rather than the often used 14 classes[4].

3.2 Baselines

To validate our approach, we compare it with general AlexNet[2] and VGG[6]. In these two baselines, we did not use target region but directly learns attributes from the whole input image. For fairness,we use the same input images and initial networks. The learning parameters are separately fine tuned for each network.

3.3 Results

Table 1 reports comparison between our network results and two baselines' results-VGG, for top-1 accuracy, and top-5 accuracy, respectively. It shows that our methods outperforms each corresponding baselines.

To demonstrate the effectiveness of target region ratio loss, we trained two more models with and without using it, respectively. The comparison are visualized in Figure 3. It shows that target regions are restored if using ratio loss. In our experiment, when applying ratio loss, the top-1 accuracy for

Table 1: Comparison of Accuracy

	AlexNet(8 layers)		*VGG-16(16 layers)*	
	Original	**Ours**	Original	**Ours**
Top-1	37.24%	**39.54%**	53.61%	**59.96%**
Top-5	68.43%	**69.48%**	80.50%	**84.14%**

Figure 3: Comparison of target region without using ratio loss(left two images) and using ratio loss(right two images)

our network based on AlexNet increases 0.78% and increases 1.47% for our network based on VGG.

Figure 4 visualizes the training log of our network based on VGG-16. The four curves in figure show the change of training loss (=*Regression Loss+Classification Loss*), validation loss (=*Regression Loss+Classification Loss*), validation dataset's accuracy of top 1 class, as well as validation dataset's accuracy of top 5 classes. The batch size for validation is 128.

At the very beginning, accuracy increases sharply with validation loss decreasing. However, after about $20,000$ iteration rounds, while the validation error increases a bit, the accuracy of validation dataset still goes up. In our analysis, we think this is because the given bounding box(Ground Truth) is not optimal for classification. Thus, the bounding box changes during learning process, which leads to higher regression loss but higher accuracy for classification. It also validates that our adaptive region generation works well, providing a more appropriate region than given bounding box.

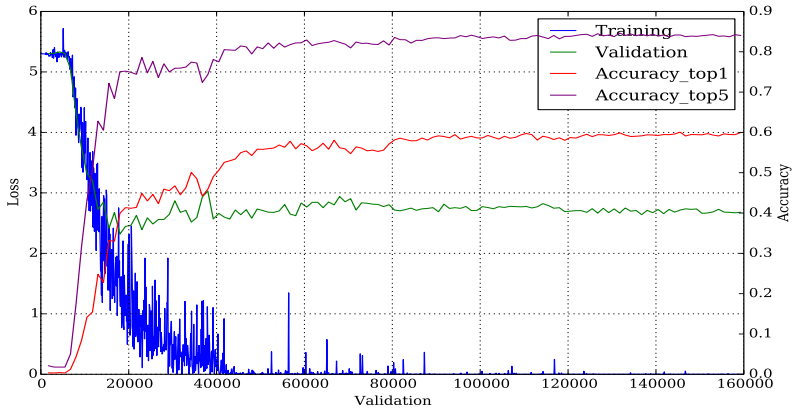


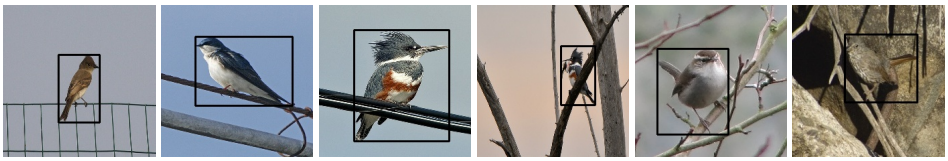
Figure 4: Curves of loss and accuracy on our network based on VGG

To illustrate the work of the adaptive target region generation, Figure 5 shows the comparison between ground truth(initial bounding box and category tag) and sample examples output from our network in testing phase.

We note that it is more likely to get wrong classification results for some images, such as the last image of Figure 5. This case generate a over-size bounding box. This is partially due to high

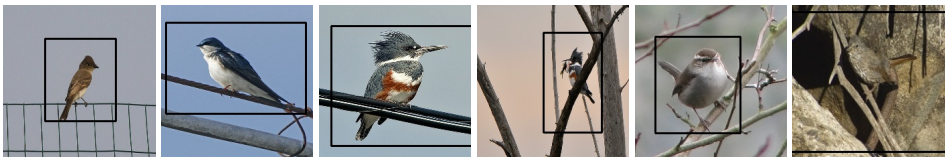
similarity between target object and its background. It leads to poor performance in prediction for target region and classification results.

Ground Truth



AF BS BK BK BW FS

Exemplar Results



AF BS BK BK BW AP

Figure 5: Illustration of adaptive target region generation.**AF = Acadian Flycatcher,BS = Bank Swallow,BK = Belted Kingfisher,BW = Bewick Wren,FS=Fox Sparrow,AP = American Pipit*

4 Conclusion

In this project we propose an end-to-end deep learning model to jointly learn adaptive target region generation and object classification. Through joint learning, indeed inspired by Spatial Transformer Network[3], the invariance of input data during training process is kept. Furthermore, noises of images are decreased due to bounding a initial target region at first. Our model is validated on CUB-200-2011 dataset via comparing its results with general AlexNet[2] and VGG-16[6].

5 References

[1]Y. LeCun, L. D. Jackel, B. Boser, J. S. Denker, H. P. Graf, I. Guyon, D. Henderson, R. E. Howard & W. Hubbard. (1989) Handwritten Digit Recognition: Applications of Neural Net Chips and Automatic Learning.*IEEE Communication, 41-46*

[2]Alex Krizhevsky, Ilya Sutskever & Geoffrey E Hinton. (2012) Imagenet classification with deep convolutional neural networks.*In Advances in neural information processing systems, pages 10971105.*

[3]Max Jaderberg,Karen Simonyan,Andrew Zisserman & Koray Kavukcuoglu. (2015) Spatial transformer networks.*In Advances in Neural Information Processing Systems, pages 20082016.*

[4]Goering C., Rodner, E., Freytag A. & Denzler, J. (2014) Nonparametric Part Transfer for Fine-grained Recognition.*IEEE Conference on Computer Vision and Pattern Recognition (CVPR).*

[5]N. Zhang, R. Farrell, F. Iandola, & T. Darrell. (2013) Deformable part descriptors for fine-grained recognition and attribute prediction.*In ICCV.*

[6]K. Simonyan, & A. Zisserman. (2015) Very Deep Convolutional Networks for Large-Scale Image Recognition.*in ICLR(oral).*

[7]Florian Schroff, Dmitry Kalenichenko, & James Philbin. (2015) Facenet: A unified embedding for face recognition and clustering. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 815823.*

[8]Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama & Trevor Darrell. (2014) Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093.*

[9]J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng& T. Darrell.(2013) DeCAF: A Deep Convolutional Activation Feature for Generic Visual Recognition. *ArXiv e-prints.*

[10]Ben Sapp & Ben Taskar. Modec.(2013): Multimodal decomposable models for human pose estimation.*In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 36743681.*