Miniplaces Challenge Report   Team TOY

Kexin Yi
Harvard University
kyi@g.harvard.edu

Aditya Thomas
MIT
adityat@mit.edu

Abstract

The following report details our approach to the Mini-Places Challenge, an assignment for the course 6.869 - Advances in Computer Vision, Fall 2017 at MIT. We explain our approach and the actual experiments conducted. We then put down our learnings from taking part in the Challenge.

1. Introduction

The combination of the human visual system and brain is quickly able to parse a scene and categorize it with respect to the objects in it and the environment [4]. Studies have shown that rather than considering scene and object representation as separate visual entities, they may best be considered as being at a similar level of abstraction [2]. To spur advances in scene recognition tasks as has happened for object recognition tasks, the Places databset has been introduced [11]. A Convolutional Neural Network (CNN) based baseline model was also included (Places-CNN) as part of the dataset [10].

In this project, we study the task of scene recognition from multiple angles and summarize our findings as follows. We first go through an overview on the motivation for the scene recognition task, the architecture of the neural network-based algorithms used for this task and then a description of the MiniPlaces Challenge which we land our focus on in this report. In Section 2, the specific approaches used to handle the scene recognition task as part of the challenge are described. In Section 3 we show the experimental results and Section 4 concludes and discusses our findings in the study.

1.1. Scene recognition

Scene recognition is an important task in computer vision, allowing the definition of a context for the object recognition task. Since scenes are composed of objects, a classification algorithm for scene recognition and classification automatically detects object descriptors within that scene [3]. Object recognition by a computer vision system include developing models for object representation and feature extraction among others [9]. It will be useful to compare competing representations and algorithms. In this respect, learning to classify scenes is an opportunity to understand the internal representation learned by a classification algorithm on a task other than object recognition [3].

1.2. Convolutional neural nets

Convolutional neural nets(CNNs) are neural networks, having units with learnable weights and biases. They differ from them in that their architecture makes the explicit assumption that their inputs will be images. Three types of layers the Convolutional layer, the Pooling layer and the Fully Connected layer are stacked to make up the CNN architecture.

The specific CNN architecture that we used is the ResNet [3] - it features special skip connections and a heavy use of batch normalization. The architecture is also missing the Fully Connected layers at the end of the network

1.3. Miniplaces challenge

The MiniPlaces Challenge tests the scene recognition capability of an algorithm. The database is a subset of the Places2 dataset [10], consisting of 100,000 images for training, 10,000 images for validation and 10,000 images for testing from 100 scene categories. All images have the same fixed size of 128 × 128.

Evaluation of the quality of a labeling will be based on the label that best matches the ground truth label for the image. Given that many environments have multiple labels (a (road) bridge could also be described as a road), the idea is to allow an algorithm to identify multiple scene categories in an image.

For each image the classification algorithm provides a top-5 predication which is compared to the gold standard.

2. Approach

2.1. Residual neural network

Residual neural nets (Resnet) are currently the most prominent model for image classification. The basic build-

Figure 1. Example images from the MiniPlaces Challenge dataset.

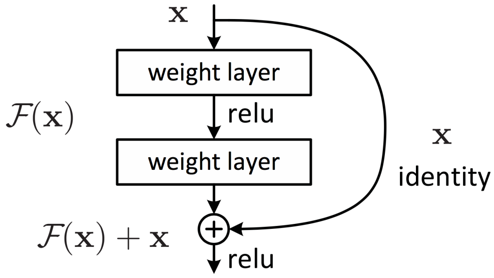

Figure 2. Schematic sketch of a residual block (figure from [3]).

ing block of resnets is a multilayer CNN block plus a residual connection that joints its input and output layer. The function that the residual block represents can be written as follows:

$$h(\mathbf{x}) = \mathbf{x} + \mathcal{F}(\mathbf{x}) \tag{1}$$

where $\mathbf{x}$ is the residual connection and $\mathcal{F}(\mathbf{x})$ is what the network really learns. Effectively the residual connection serves as a regularizor for the network so that the model doesn't need to learn the identity map as it comes for free. This nice property enables the network to go very deep without suffering from overfitting, and therefore possesses stronger representation power. In this project we are going to adopt Resnet as our main architecture. Other than image classification, Resnets are also widely applied in many other deep learning domains such as neural translation [8] and deep reinforcement learning for game playing [6].

2.2. Edge detection and Gaussian filtering

Edge detection is a method of determining points in an image where there are discontinuities. The importance of edge detection is from the intuition that points in the image where there are discontinuities i.e. edges are where there is a significant change in depth, orientation, reflectance or illumination and that in turn are indicators of the edge of an object [7].

We pre-processed the images using an object detection algorithm [5]. We used two 2-D Gaussian filters of dimensions 7*7 and 13*13 as smoothing filters. This was followed by detecting the zero-crossings using a 3*3 Laplacian kernel.

2.3. Ensemble method

In machine learning, it is common to combine results from multiple models and use a (weighted) vote to determine the overall output. This is often referred to as the "ensemble method". For our scene classification task, the ensemble method can be formulated as follows: given a bunch of individual classifiers $M1, M2, ...M_K$ that outputs logits $l_1, l_2, ...l_K$, the final class prediction is based on the weighted average of the individual scores

$$l_F = \sum_i w_i l_i \tag{2}$$

where the weights w_i are normalized $\sum_i w_i$ = 1. The probability for each class is then given by the softmax function

$$p_k = \text{softmax}_k(l_F). \tag{3}$$

In general, the optimal set of weights w_i^* can be computed by an iterative process like the Adaboost algorithm [1]. In this project, we simplify this process by performing grid search over the weight space. The final outputs from the ensemble method are computed under the optimal weights verified on the validation set.

3. Experiments

In this section we study the effect of various aspects on the scene recognition performance, including the depth of network, edge enhancement, and ensemble method. Our best performing model so-far has 0.239 top-5 error and 0.546 top-1 error on the test set. All experiments are implemented in Pytorch. For all models trained, we choose a batch size of 100 and use Adam for gradient based optimization with learning rate equal to 0.001. More details can be found in the submitted code.

3.1. Network depth

One major advantage of Resnet is its ability to gain more representation power by going deeper [3]. It is often believed that the performance of Resnet always increases with depth. In this part we study how this factor affects the performance on the scene recognition task. We train Resnet models with various depths (18, 34, 50, 101) on normalized image inputs. Each model is trained for 15 epochs over the miniplaces training set and the accuracy is evaluated on the validation set twice for every epoch. Over the entire training process, the best performing model on top-5 validation accuracy is recorded and used for prediction.

Figure 3 shows the validation accuracy of all models over the training process, which indicates an unrecognizable performance across models with different depth on

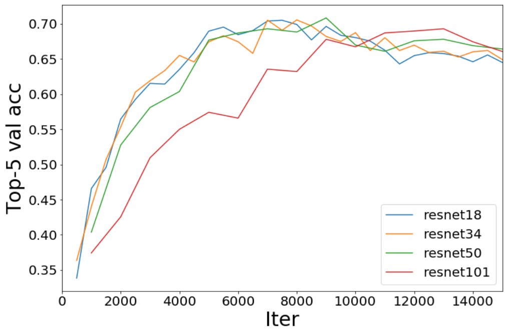

Figure 3. Top-5 validation accuracy versus iteration for Resnet models with various depth.

both learning speed and accuracy. The best validation accuracy achieved by all models are close to 0.7. After 8000 iterations, validation accuracy starts to go down for all depth and the neural nets become overfitted. We also run Resnet34 for more than 60 epochs to make sure the model is not trapped at local minimum over the first 15 epochs. The similar behavior on all models shows that the network depths is not a key factor for scene recognition. More discussions on the result will be made in section 4.

3.2. Edge-enhancement

To perform edge enhancement on the images, we add a preprocessing module to our training and evaluation pipeline. Specifically, we apply on each image a 13 by 13 Gaussian filter with standard deviation equal to 2 for each color channel. We then add those extracted edges back to the original image as an enhancement signal. The preprocessed images are trained and evaluated on Resnet34 with the same set of parameter as in section 3.1.

The result for both top-5 and top-1 validation accuracy can be found in figure 4. Unfortunately we did not see an improvement in performance but rather a decrease in accuracy due to information loss on the Gaussian filters. This result, along with the result from section 3.1, further demonstrates that the dominating factors for scene recognition are the long range visual features instead of local ones.

3.3. Ensemble method

Finally, we form an ensemble model as described in section 2.3 by gathering all models we have trained: 1, Resnet34; 2, Resnet34 with edge-enhancement; 3, Resnet18; 4, Resnet50 and 5, Resnet101. To find the optimal weights, we run a grid search over $w_{1:5}$ and pick the set of parameters with highest validation accuracy. To our surprise the ensemble method boosts up the validation accuracy by about 6% in top-5 accuracy. Even though the individual changes we applied to the baseline Resnet model do not yield obvious improvements, by putting them together we are able to achieve a notably better result. The best recognition accuracy (0.239 top-5 error and 0.546 top-1 error) we are able to achieve so far comes out from the particular set up where $w_1 = 0.28, w_2 = 0.22, w_3 = 0.13, w_4 = 0.19, w_5 = 0.18$. Comparison between the performance of all individual models we have studied and the ensemble model are summarized in table 1.

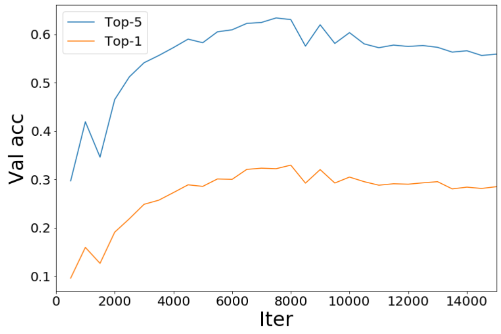

Figure 4. Validation accuracy versus iteration for edge-enhanced images.

Model	Top-5 accuracy	Top-1 accuracy
Resnet18	0.7050	0.3982
Resnet34	0.7055	0.4009
Resnet34(eh)	0.6335	0.3250
Resnet50	0.7083	0.4004
Resnet101	0.6929	0.3844
Ensemble model	**0.7686**	**0.4619**

Table 1. Best validation accuracy over the first 15 training epochs.

4. Conclusions and discussions

The key findings of our study can be summarized as follows: first of all, adding more depth to the network does not contribute to the overall performance. Unlike single object classification, scene recognition is based on larger scale visual features. This is probably the reason why one cannot get better performance by pushing forward the depth of the model nor can we get any improvement by enhancing local features. On the other hand, the striking improvement from putting together multiple models shows that the bottleneck of this task is really the width of the model not the depth. Therefore, one immediate modification it suggests is to increase the width of the pipeline. Specifically, we could use a pyramid-like structure on which the transversal size of the feature maps decreases slowly through the layers. Moreover, semantical features and interactions between scene objects are also highly correlated to the scene category (for example, if there are people sitting at a dinning table, it is very likely that the scene is a restaurant). Extracting and utilizing those object level features will be a good direction for future studies.

acknowledgement

We wish to put on record our thanks to Chollette Olisah, who is a listener of 6.869, for valuable discussions and guidance, including the idea of image enhancement and implementing the image preprocessing module.

References

[1] M. Collins, R. E. Schapire, and Y. Singer. Logistic regression, adaboost and bregman distances. *Machine Learning*, 48(1):253–285, 2002.

[2] L. Fei-Fei, R. VanRullen, C. Koch, and P. Perona. Why does natural scene categorization require little attention? exploring attentional requirements for natural and synthetic stimuli. *Visual Cognition*, 12(6):893–924, 2005.

[3] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[4] T. Konkle, T. F. Brady, G. A. Alvarez, and A. Oliva. Scene memory is more detailed than you think: The role of categories in visual long-term memory. *Psychological Science*, 21(11):1551–1556, 2010.

[5] C. C. Olisah, S. Nunoo, P. Ofedebe, and G. Sulong. Expressing facial structure and appearance information in frequency domain for face recognition. *arXiv preprint arXiv:1704.08949*, 2017.

[6] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton, et al. Mastering the game of go without human knowledge. *Nature*, 550(7676):354–359, 2017.

[7] V. Torre and T. A. Poggio. On edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (2):147–163, 1986.

[8] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, et al. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*, 2016.

[9] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba. Object detectors emerge in deep scene cnns. *arXiv preprint arXiv:1412.6856*, 2014.

[10] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[11] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva. Learning deep features for scene recognition using places database. In *Advances in neural information processing systems*, pages 487–495, 2014.