# Recognition of Tourist Attractions

Yuanfang Li (yli03@stanford.edu)
Xin Li (xinli16@stanford.edu)
Chang Yue (changyue@stanford.edu)

## I Introduction

With the mass of images we are presented with everyday, there are many times when we see a photo of a place that we would like to visit but are unable to determine where it is. The ability to recognize landmarks from images can be extremely useful both when choosing a travel destination and when trying to identify landmarks in a foreign place. Our project aims to be able to recognize the specific location using machine learning methods. Due to time and complexity constraints, we limit ourselves to recognizing ten famous tourist attractions in Beijing. The input to our algorithm is an image. We then use a convolutional neural network to extract image features and an SVM which outputs the predicted attraction.

We use the following ten attractions in our model: Tiananmen Square, the Forbidden City, Yuanmingyuan Park, Beihai Park, Beijing National Stadium (Bird Nest), Beijing National Aquatics Centre (Water Cube), CCTV Headquarters, the Great Wall, National Centre for the Performing Arts (Bird Egg) and Fragrant Hills Park. These attractions were chosen because they each have distinct features that make the classification task easier and are well known enough to amass a large dataset for training.

## II Related Works

While there are no pre-existing works on tourist attraction classification, Zhou et al[1] focus on using convolutional neural networks for scene recognition, which aligns closely with our project. Convolutional neural networks are particularly suited for extracting image features. Earlier layers extract general features such as edges while later layers are finetuned to extract features specific to the dataset and classification problem.[2]

The Places-CNN uses the same network structure as AlexNet[3] but is trained on a dataset of scenery images from the Places database as opposed to object images from ImageNet. A comparison of the features extracted by AlexNet vs Places-CNN shows that the two networks start to diverge after the first convolution layer due to the differences in training images. Furthermore, Places-CNN was able to achieve test accuracy of 50.0% on a database of 205 scenery images while AlexNet with an additional SVM classifier was only able to achieve 40.8% test accuracy.[1]

Based on these results, Places-CNN should map well to our application since tourist attractions fall into the scenery rather than object category, so we can apply transfer learning rather than training the neural net from scratch. Thus, we choose to train our model by first using Places-CNN as a fixed feature extractor and then train a classifier on the extracted features.[4]

For classification tasks, most of the fully-connected and convolutional neural networks employ the softmax fuction for the final layer. Tang [5] demonstrated a small but consistent advantage of replacing the softmax layer with a linear support vector machine. Results using L2-SVMs (DLSVM) showed better performance on some of the most popular deep

learning datasets such as MNIST, CIFAR-10 and the ICML 2013 Representation Learning Workshop's face expression recognition challenge.

Thus, after extracting the 4096 features from the last fully-connected layer of PlacesCNN, we trained both a softmax classfier and SVM classifier for the final layer. Then we compared the performance of these two methods to verify if multiclass SVM does work better than softmax on our dataset.

## III Data and Feature Extraction

We created our own dataset of 10000 images and further split this into 7000 training images, 2000 validation images and 1000 test images. These images were obtained through Google image search by modifying a script to download the images returned.

To ensure our dataset included images of each attraction from different directions and in different weather conditions, we varied the key terms in our search. We downloaded a total of 1000 images for each category, then manually went through the images to remove any irrelevant results. We further augmented the data by cropping and horizontally reflecting to achieve 1000 images per category.

We extracted two sets of features from each image:

### *1. Places-CNN*

Each image was fed through the Places-CNN model, pre-trained on 2,448,873 images from 205 scene classes. The model parameters were obtained from the caffemodel[6] and converted into Tensorflow. We extracted the size 4096 output from the fc7 layer to use as the input features for our classifier.

### *2. Pixel Ratio*

After examining our raw images and the features extracted by Places-CNN, we found differences between day and night images and features of the same attraction. Initial training of the classifier also showed a drop in accuracy when the dataset contained both day and night images, thus we decided to train separate classifiers for day and night and use an additional binary classifier to determine which model to use at test time.

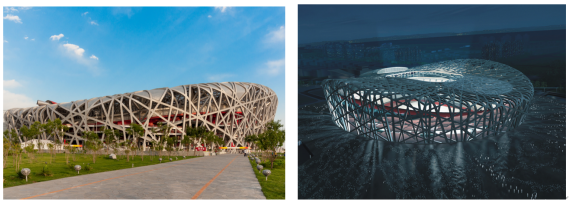

Figure 1: Bird Nest Stadium, day vs. night

To do this, we computed the pixel ratio of the image by counting the total number of pixels of each value and then dividing by the total number of pixels. Accounting for the entire RGB colour space, the feature vector size would be $2^{8^3}$, so we further mapped the $2^8$ possible pixel values of each colour to 4 values. This gives a feature vector of size $4^3 = 64$, which we used as the input to our day vs. night classifier. We based this feature extraction method on similar work by Cardone [7].

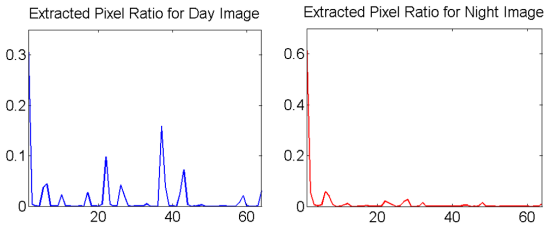


Figure 2: Day vs. night pixel ratio

## IV Method

### *1. Softmax*

$\sigma(z)_j = \frac{\exp(z_j)}{\sum_{k=1}^{K} \exp(z_k)}$, the softmax function, outputs a value between 0 and 1 that can be interpreted as the probability of sample z being from class j. We first trained a softmax classifier with K=10 classes, using stochastic gradient descent with momentum to minimize the cross-entropy[8]:

$$J(\theta) = -[\sum_{i=1}^{m}\sum_{k=1}^{K} \mathbb{1}\{y^{(i)=k}\} \log \frac{\exp(\theta^{(k)T}x^{(i)})}{\sum_{k=1}^{K}\exp(\theta^{(j)T}x^{(i)})}]$$

$$\Delta\theta := \eta\nabla J(\theta) + \alpha\Delta\theta$$

$$\theta := \theta - \Delta\theta$$

Momentum was used in order to minimize oscillations and accelerate the training process.[9] Training was run for 10,000 epochs using a grid search with learning rates of 0.01, 0.001 and 0.0001 and momentum equal to 0.1, 0.5 and 0.9. We used a batch size of 100 images.

Based on the validation accuracy, we determined the best model occurred with learning rate = 0.01 and momentum = 0.9. We used early stopping to prevent overfitting by calculating validation error every 10th epoch and stopping training when no decrease in error was found after 100 epochs.[9]

*2. SVM*

When examining our day and night images, we found that some day images were taken at dusk/dawn and some night images contained attractions that were very well lit. As a result, there was some overlap in pixel ratio features between day and night images, which would make it difficult to linearly separate them. Thus we chose to use an SVM to classify the images as either day or night.

Unlike softmax, the SVM is non-probabilistic and seeks to fit a hyperplane that will separate two classes with as large a distance/margin as possible. However, by using a soft-margin SVM we can still train a reasonable classifier even when the data is not linearly separable. The loss to be minimized in this case is[8]:

$$J(\theta) = \frac{1}{n}\sum_{i=1}^{m}\max(0, 1 - y_i(\theta^T x_i)) + \lambda||\theta||^2$$

We used the same training method as for softmax and also performed a grid search to find optimal value of the regularization hyperparameter, $\lambda$, which was 0.01.

We also trained a multiclass SVM with 10 classes for our attractions classifier using the one vs. all approach[8], which fits a hyperplane that separates each class from all others. The output of the model is the class for which the distance from the hyperplane is greatest.

Based on the validation accuracy of the two models, we determined that multiclass SVMs are more suited for our classification task than a softmax classifier. Our full classification pipeline is presented in Figure 3.

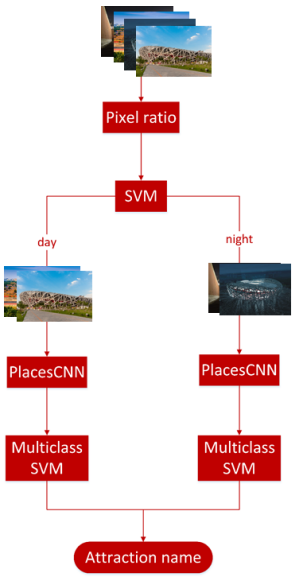


Figure 3: Complete classification pipeline

## V Results

| | Training | | Validation | |
|---|---|---|---|---|
| | Data | Accuracy | Data | Accuracy |
| Day/night SVM | 2800 | 96.0% | 1200 | 94.0% |
| Day softmax | 5600 | 52.3% | 1400 | 46.6% |
| Day SVM | | 92.3% | | 83.9% |
| Night softmax | 1400 | 82.2% | 600 | 75.2% |
| Night SVM | | 95.5% | | 87.9% |
| | **Test Data** | | **Test Accuracy** | |
| **Complete** | **1000** | | **84.1%** | |

Table 1: Accuracy of separate classifiers

Table 1 shows the accuracy results of the different classifiers we trained. Whereas Tang [5] demonstrated a small difference between softmax and SVM classifier, here the multiclass SVM classifier performed significantly better than the softmax classifier. This is

likely because the different classes were not completely linearly separable, which we anticipated would be the case. Although most of the attractions we chose are very distinctive, large locations like Beihai Park have a variety of images, many of which share similar features to other attractions. Furthermore, since Places-CNN was trained on largely natural scenes and most of our attractions contain manmade components, the features extracted from the image many not have been the best ones for our application.

Comparison of the day and night classifiers shows that the night classifier is more accurate, particularly for the softmax model. Since most of the attractions are brightly lit against a dark background at night, noise from the surroundings is not as prominent and only the attractions themselves are visible, so the classes are more separable. Figure 4 shows the Water Cube at day and night and the features extracted by Places-CNN. In the day image, skyscrapers are visible in the background while at night only the Water Cube is visible. From the plot of extracted features, it can be seen that i) day and night features are indeed different and ii) the magnitude of non-zero night features is much higher.

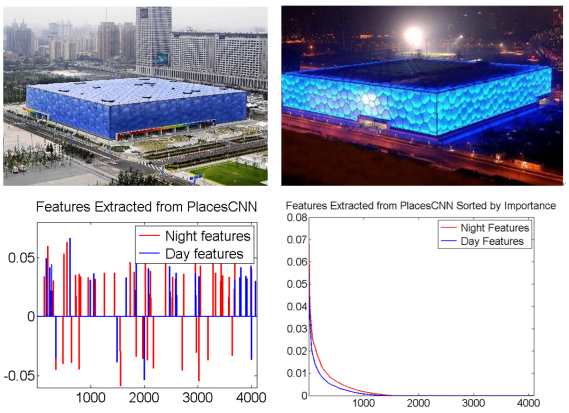


Figure 4: Day vs. night feature importance

By examining Table 2, we see that results are better for venues, such as Bird Egg, CCTV and Water Cube, which are known for their eclectic architecture. Conversely, the classifier performs poorly on attractions like Beihai Park, Fragrant Hill and the Forbidden City, which are more spread out and generic looking.

From the test results, recall for Beihai Park is significantly lower. We examined the incorrectly predicted images and found that many of the images were taken in dim light. This caused the day vs. night classifier to incorrectly label them as night images and feed them through the wrong multiclass SVM.

| | Precision | Recall |
|---|---|---|
| Beihai Park (C0) | 89.6% | 69.0% |
| Bird Egg (C1) | 97.8% | 91.0% |
| Bird Nest (C2) | 81.0% | 85.0% |
| CCTV (C3) | 91.0% | 91.0% |
| Forbidden City (C4) | 72.7% | 80.0% |
| Fragrant Hill (C5) | 75.9% | 85.0% |
| Great Wall (C6) | 87.4% | 83.0% |
| Tiananmen (C7) | 79.0% | 79.0% |
| Water Cube (C8) | 82.7% | 91.0% |
| Yuanmingyuan Park (C9) | 87.9% | 87.0% |
| **Total Accuracy** | | **84.1**% |

Table 2: Precision and recall

Furthermore, the confusion matrix in Table 3 shows that many images of Beihai Park were labelled as the Forbidden City. Figure 5 shows an image of Beihai Park that was classified as the Forbidden City and an actual image of Forbidden City. We can see that they are indeed very similar.

Figure 5: Beihai Park (left) vs. Forbidden City (right)

Both precision and recall for Bird Egg are very high. This is due to the fact that it is the only attraction with a distinctly curved shape - all other attractions either have angular shapes or are natural landscapes/parks.

| | C0 | C1 | C2 | C3 | C4 | C5 | C6 | C7 | C8 | C9 |
|---|---|---|---|---|---|---|---|---|---|---|
| **C0** | 69 | 0 | 0 | 0 | 14 | 5 | 0 | 5 | 0 | 7 |
| **C1** | 0 | 91 | 3 | 0 | 0 | 1 | 0 | 4 | 0 | 1 |
| **C2** | 2 | 0 | 85 | 2 | 0 | 4 | 0 | 0 | 9 | 0 |
| **C3** | 0 | 1 | 1 | 91 | 2 | 0 | 2 | 2 | 1 | 0 |
| **C4** | 2 | 0 | 0 | 3 | 80 | 3 | 7 | 2 | 2 | 1 |
| **C5** | 1 | 0 | 9 | 0 | 1 | 85 | 1 | 2 | 1 | 0 |
| **C6** | 0 | 1 | 2 | 0 | 2 | 12 | 83 | 0 | 0 | 0 |
| **C7** | 1 | 0 | 2 | 0 | 10 | 1 | 0 | 79 | 5 | 2 |
| **C8** | 0 | 0 | 3 | 0 | 1 | 1 | 1 | 2 | 91 | 1 |
| **C9** | 2 | 0 | 1 | 4 | 0 | 0 | 1 | 4 | 1 | 87 |

Table 3: Confusion matrix

## VI Conclusion

The multiclass SVM model performed significantly better than the softmax classifier, giving a final test accuracy of 84.1% for the complete classifier. This is likely due to the fact that features extracted by Places-CNN are not the exact features we require and so are difficult for the softmax classifier to linearly separate.

Based on the results from Zhou et al[1], networks trained on different datasets will differ more in deeper layers. Divergence between the models is particularly evident from the second convolution layer onwards. Given more time and computing resources, we would like to extract the image features from the output of the first convolution layer and then train the remaining layers on our dataset.

Ultimately, we would like to be able to extend our classifier to more than just ten attractions. Since the attractions we chose are very well known to begin with, the usefulness of our classifier is limited. Using less well known attractions may provide difficulties in obtaining a large enough dataset but will be more useful in achieving the goal of allowing people to choose their travel destinations based on the photos they see.

## References


[1] 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*, 2014, pp. 487–495.

[2] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *European Conference on Computer Vision*. Springer, 2014, pp. 818–833.

[3] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[4] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on knowledge and data engineering*, vol. 22, no. 10, pp. 1345–1359, 2010.

[5] Y. Tang, "Deep learning using linear support vector machines," *arXiv preprint arXiv:1306.0239*, 2013.

[6] M. C. Science and A. I. Laboratory. Places CNN. [Online]. Available: http://places.csail.mit.edu/downloadCNN.html

[7] G. Cardone. Day and night: an image classifier with scikit-learn. [Online]. Available: http://www.ippatsuman.com/

[8] K. Duan, S. S. Keerthi, W. Chu, S. K. Shevade, and A. N. Poo, "Multi-category classification by soft-max combination of binary classifiers," in *International Workshop on Multiple Classifier Systems*. Springer, 2003, pp. 125–134.

[9] S. Ruder, "An overview of gradient descent optimization algorithms," *arXiv preprint arXiv:1609.04747*, 2016.