

# Observer Localization using ConvNets

(Stanford CS231n class project, Winter 2016)

Alon Kipnis  
Department of Electrical Engineering  
Stanford University  
350 Serra Mall, Stanford, CA  
kipnisal@stanford.edu

## 1. Introduction

Tracking and retrieving the exact location of an object in a 3D environment is a major challenge in many situations such as navigation and orientation inside an indoor environment. Current radio based localization techniques such as Global Positioning Systems (GPS) and cellular network data are limited by their accuracies due to relatively large wavelength of radio signals compared to accuracies required in many of these scenarios.

In this project we explore the possibility of detecting the location of an object in a scene, or alternately, the location of the observer based on visual data. Visual data is indeed the default way humans orient themselves in an indoor environment is by visual data. In addition we propose to use 3D models of real objects to obtain training data. Below are a few examples where such an approach to localization can be applied:

- Drone or autonomous navigation – the path can be learned offline based on 3D model. The machine loads a pre-trained ConvNet according to the particular path it chooses.
- Localizing an object in a 3D environment may assist 3D printing tasks and other indoor location tasks where precision required is smaller than radio wavelength.
- Since 3D model emphasizes the outline of an object rather than its textures, it is suggested that training based on 3D models will experience less over-fitting in classifying objects which has different textures (e.g. houses, cars, airplanes, trucks, cups, bananas).
- Fine-tuning of navigation and motoric tasks such as 3D printing, manufacturing and hovering [1].

### 1.1. Related Works

Geolocalization using visual data has been considered in many different settings. Works on geo-localizations can be

roughly divided into two according to the resolution of the localization: (1) large scale – the geographic accuracy required is of the order of several kilometers, such as in recognizing cities and street [2, 3, 4]. (2) small scale – accuracy required is within a few meters or even less. The problem considered in this project related to the second class, i.e., small scale and high resolution geo-localization.

Works which consider similar settings to ours can be found in [5] and the references therein. Localization in scenes based on their 3D models was considered in [6] but in with an approach which is not based on ConvNets. We note that a geo-localization problem is significantly different than object localization problem [7], since the former has a ground truth which is based on spatial data measured from real world scenes while the ground truth of the latter only depends on a single sample image.

### 1.2. Contribution

In this project we consider the task of localizing an observer in a scene from a single real-time picture in a supervised learning setting using ConvNets. The scene is first learned by a ConvNet trained to classify the angle into 12 uniform sections over the circle. The testing is performed by examining the classification result on a previously unseen sample of the *same* scene.

In order to get a large amount of labeled data we use a 3D modeling software. We render a given scene multiple times from different locations of the camera. For each image we record the angle of the camera with respect to the center of the scene.

Finally, we explore the possibility of predicting the camera location of real world scenes based on their 3D model.

### 1.3. Organization

The rest of this report is organized as follows: In Section 2 we formalize the problem of camera/viewer localization and explain the fidelity criterion used for evaluation. In

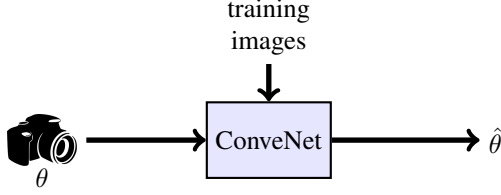


Figure 1. Camera/observer relative angle detection problem setting.

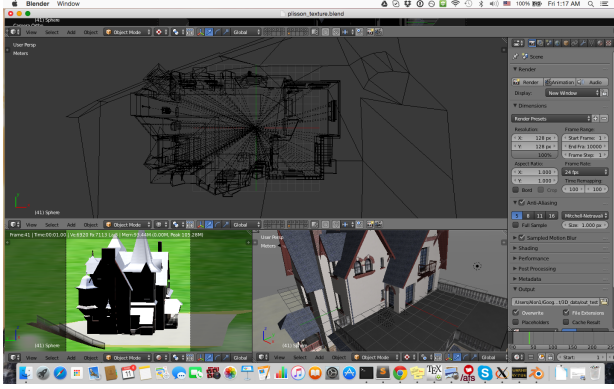


Figure 2. GUI of 3D modeling python package Blender. The data is generated by rendering multiple images of the model from different camera locations.

Section 3 we explain the technical approach taken as well as the structure of the ConvNet. The results are presented and discussed in Section 4. Concluding remarks are given in Section 5.

## 2. Problem Statement

We consider a geolocation problem as depicted in Figure 1. In this problem, the network is required to retrieve the spatial location of the camera given an input image. We formulate task problem as a classification problem by dividing the circle into 12 identical sections. The goal of the network is to produce an estimate of the particular class of the viewer/camera angle. The loss in this case is indicator loss (also known as the Hamming loss or Kronecker delta):

$$\ell_I = \mathbf{1}_{(c_{\text{true}}=\hat{c})}.$$

The classification *accuracy* is defined as the average of  $\ell_I$  loss over the data set. Namely, the number of correct classifications divided by the number of input images.

## 3. Technical Approach

### 3.1. Data

The data is obtained using a 3D modeling software Blender (GUI for the Python Blender package, Figure 2).

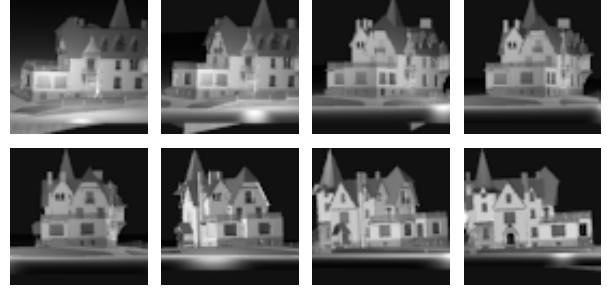


Figure 3. Example dataset1: manor house

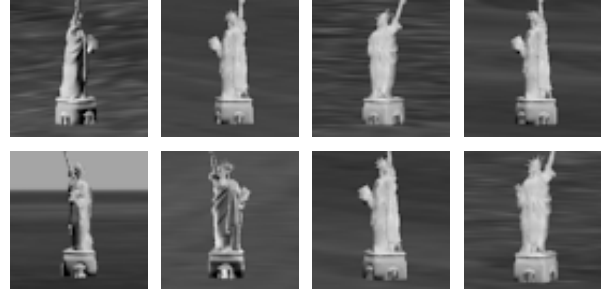


Figure 4. Example dataset2: Statue of Liberty

For a particular scene or object, multiple images are generated that corresponds to different angles of the camera with respect to the object located at the center of the scene. These angles are stored as the labels of the training data. Examples for images from the training data generated by this procedure are given in Figures 3 and 4.

The *manor house* example includes 9000 labeled images splitted into training, validation and testing data according to the ratio 8 : 1 : 1. The *Statue of Liberty* example includes 5000 labeled images splitted according to the same ratio.

### 3.2. Network Architecture

We use the ConvNet architecture provided in Figure 5 which is based on the AlexNet [8]. The network consists of three convolution-ReLU-maxpool-batchnorm-dropouts blocks, followed by 2 fully connected layers of 1024 neurons. The number of outputs at the last fully connected layer is 12, which corresponds to different intervals of the circle.

We realized the ConvNet illustrated in Figure 5 using the scientific computing framework TensorFlow<sup>TM</sup>. The TensorFlow computational graph that realizes data reading, training and validation is illustrated in Figure 6.

### 3.3. Loss Function

The loss function is log-loss or cross entropy loss between the distribution that gives 1 to the true angle class

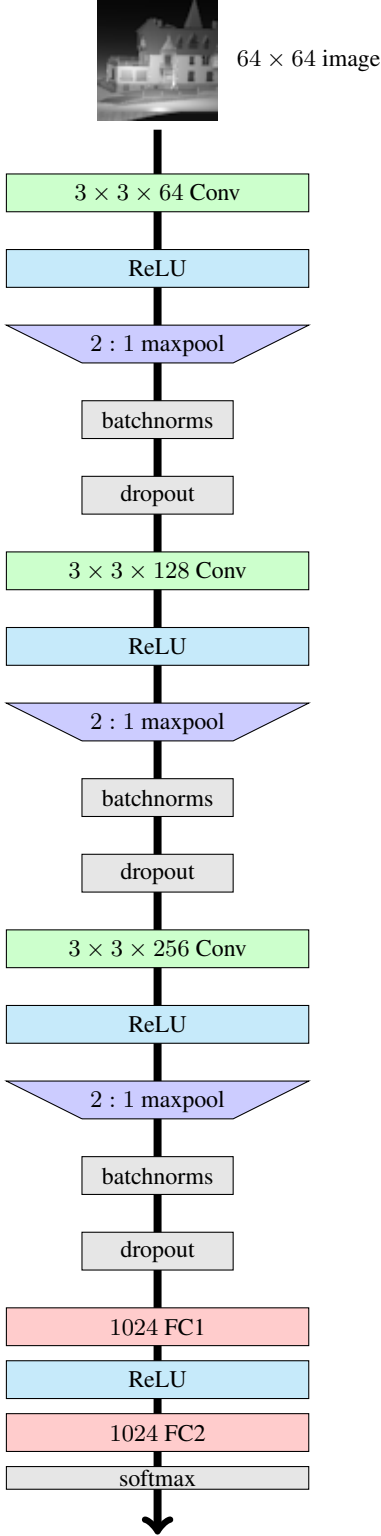


Figure 5. ConvNet architecture

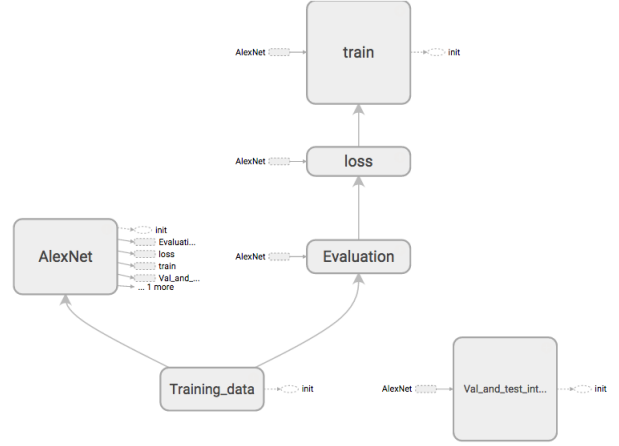


Figure 6. Tensorboard realization of the training and evaluation model. Note that this graph also includes data reading interfaces for training and evaluation.

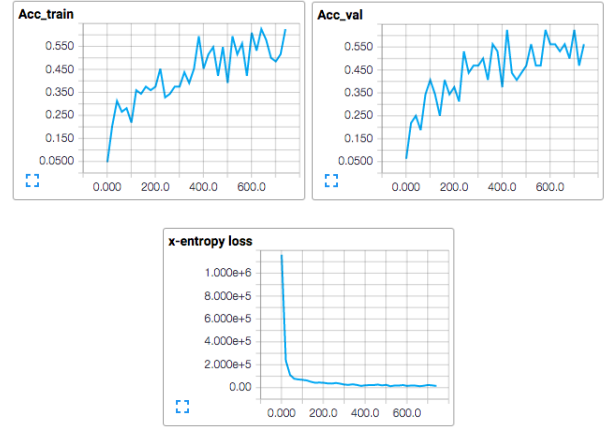


Figure 7. TensorBoard statistics for the Statue of Liberty dataset.

and zero otherwise. For a single sample this is defined by

$$\ell_i = -\log(\hat{p}_i), \quad (1)$$

where  $\hat{p}_i$  is the probability that the network assigns to the true class. The actual loss is the mean of  $\ell_i$  over a batch of samples. In addition, we added a regularization term to (1) which is the  $L_2$  norm of the weights in the last two fully connected layers (FC1 and FC2 in Figure 5).

### 3.4. Learning Process

The result of the learning process for the manor house example and the Statue of Liberty example are given in Figure 7.

With the manor house data the network achieved 100% accuracy on the training and about 99% on the validation set. With the Statue of Liberty data the network achieved 66% on training set and 64% on the validation set. Train-

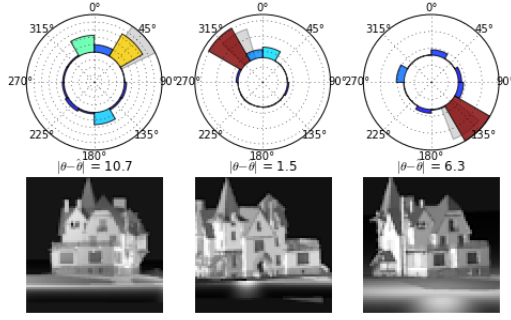


Figure 8. Three examples for angle-class predictions with respect to test data for the manor house example. The probability assigned by the network to each direction is given by the different colors (small probabilities are given cold colors) The class that contains the true angle is painted in gray. The absolute error between the true angle and the predicted angle (the class center) is also provided.

ing statistics of the Statue of Liberty example from TensorBoard are provided in Figure 7.

## 4. Results

Training, validation and testing scores with respect to the two models are given in the table below.

	train	validation	test
Manor	100%	99%	96%
Liberty	66%	64%	56%

Example for the output probabilities predicted by the model on the test set and the corresponding ground truth are illustrated in Figures 8 and 9.

### 4.1. Discussion

The very high classification accuracy obtained in the manor house example suggests that our angle detection task is not too challenging for a ConvNet considered with such a high number of training samples. Nevertheless, these results shows that viewer angle detection from visual data is possible and motivate us to explore harder geolocalization tasks in the future.

The difference in training performances between the two models may be due to the size of the data (9000 vs 5000 samples) and other properties of the object. For example, excluding her hand, low resolution images of Lady Liberty seem to have more radial symmetry which make angle detection harder. This also explains the difference in the test performance.

Another way to evaluate the classification performance is by considering the absolute error between the true direction

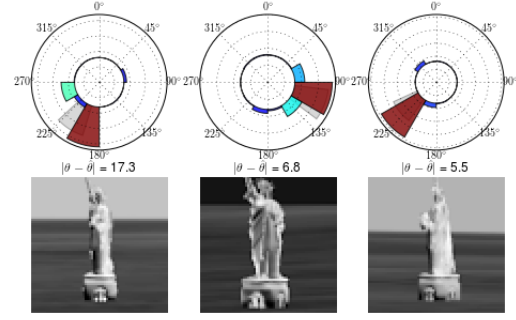


Figure 9. Three examples for angle-class predictions with respect to test data for the Statue of Liberty example. The probability assigned by the network to each direction is given by the different colors (small probabilities are given cold colors) The class that contains the true angle is painted in gray. The absolute error between the true angle and the predicted angle (the class center) is also provided.

and the direction predicted by the network. The rational is that even when a classification error occurs, the actual angle may still be close to the angle predicted by the network (the center of the interval of the predicted class). The absolute error between the true angle and the angle predicted by the network are also given in Figures 8 and 9 for individual examples.

### 4.2. Real data Classification Based on 3D Model

Once the network is trained on the 3D model of the Statue of Liberty, we examined its performance with respect to estimating the camera angle of real images of the Statue of Liberty taken from the internet. An example for the results are given in Figure 10.

It can be seen that in most cases the system provides high probability to what seems to be the true angle from which the picture was taken. We can speculate the inaccuracy of the real data compared to the test data of the trained 3D model is due to lighting, texture or background conditions of the real data which are significantly different than the rendered 3D model. It would be beneficial to render the 3D model under different sky lighting condition in order to improve prediction on real images.

### 4.3. Convolution-Layer Activations

Illustrating the activation from different layers in the trained network would help understand the features learned by the network in order to predict the camera angle [9]. Examples for these activations are given in Figures 11 and 12 for the manor house model and the Statue of Liberty model, respectively. Note that the two networks were trained separately, so comparing both Figures feature-by-feature is meaningless.

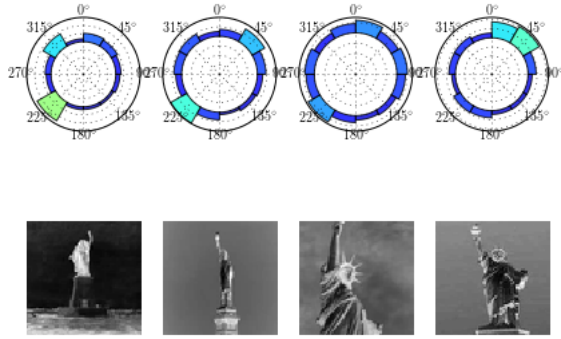


Figure 10. Camera angle prediction result on real images of the Statue of Liberty.

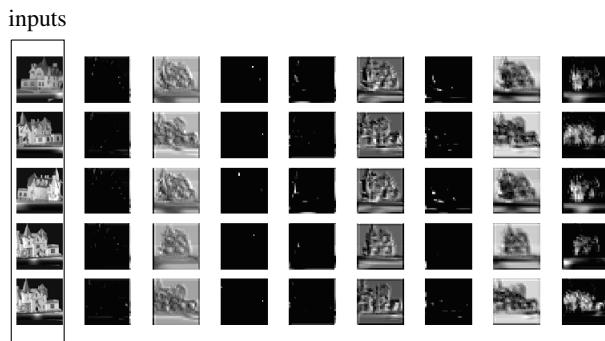


Figure 11. Activation maps for five input images from second convolution layer in the manor house example.



Figure 12. Activation maps for five input images from second convolution layer in the Statue of Liberty example.

It can be observed that areas of bright light and edges are emphasized by this convolution layer.

## 5. Conclusions

We have considered a geolocation task of estimating the angle of the viewer/camera with respect to an object. We trained a ConvNet with architecture similar to AlexNet to predict the angle in a classification model of 12 different angles. The data is obtained from a 3D model of the ob-

ject using a 3D modeling program (Blender). The computation graph was built and trained using TensorFlow™ framework. We trained and tested the network over two datasets of 9000 samples and 5000 samples, respectively.

The trained network achieves very high classification accuracy on the test data (97%) for the model with 9000 samples and relatively high accuracy (56%) for the dataset with 5000 images. These results suggest that ConveNets can be successfully trained to retrieve angle and possibly other location parameters from visual data. This fact motivates the study of more complicated geolocation setting. For example, it suggests that ConvNets can be trained to localize an observer/camera inside a closed room or building.

Another important questions which was raised in this project is the performance of classification of real word images trained on a data taken from a 3D model of the real scene/object. Unfortunately, we have failed to present positive results in this direction: our ConveNet trained on a 3D model of the Statue of Liberty did not provide meaningful predictions for real images of the Statue.

## References

- [1] T. Wu, F. Huang, J. L. Yao, and B. Yang, “Geo-localization based scene recognition,” in *Multimedia Big Data (BigMM)*, 2015 IEEE International Conference on, April 2015, pp. 244–247.
- [2] G. Baatz, O. Saurer, K. Köser, and M. Pollefeys, “Large scale visual geo-localization of images in mountainous terrain,” in *Computer Vision–ECCV 2012*. Springer, 2012, pp. 517–530.
- [3] T.-Y. Lin, Y. Cui, S. Belongie, and J. Hays, “Learning deep representations for ground-to-aerial geolocation,” in *Computer Vision and Pattern Recognition (CVPR)*, 2015 IEEE Conference on. IEEE, 2015, pp. 5007–5015.
- [4] A. V. Peddada and J. Hong, “Geo-location estimation with convolutional neural networks,” 2016. [Online]. Available: [http://cs231n.stanford.edu/reports/CS231N\\_Final\\_Report\\_amanivp\\_jamesh93.pdf](http://cs231n.stanford.edu/reports/CS231N_Final_Report_amanivp_jamesh93.pdf)
- [5] M. Aubry, B. C. Russell, and J. Sivic, “Visual geo-localization of non-photographic depictions via 2d-3d alignment,” *Visual Analysis and Geolocation of Large-Scale Imagery*, 2015.
- [6] A. R. Zamir and M. Shah, “Image geo-localization based on multiplanearest neighbor feature matching using generalized graphs,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 8, pp. 1546–1558, Aug 2014.

- [7] J. Tompson, R. Goroshin, A. Jain, Y. LeCun, and C. Bregler, "Efficient object localization using convolutional networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 648–656.
- [8] 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.
- [9] 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," *CoRR*, vol. abs/1310.1531, 2013. [Online]. Available: <http://arxiv.org/abs/1310.1531>