PlaNet - Photo Geolocation with Convolutional Neural Networks

Tobias Weyand[1(✉)], Ilya Kostrikov[2], and James Philbin[3]

[1] Google, Los Angeles, USA
weyand@google.com
[2] RWTH Aachen University, Aachen, Germany
ilya.kostrikov@rwth-aachen.de
[3] Zoox, Menlo Park, USA
philbinj@gmail.com

Abstract. Is it possible to determine the location of a photo from just its pixels? While the general problem seems exceptionally difficult, photos often contain cues such as landmarks, weather patterns, vegetation, road markings, or architectural details, which in combination allow to infer where the photo was taken. Previously, this problem has been approached using image retrieval methods. In contrast, we pose the problem as one of classification by subdividing the surface of the earth into thousands of multi-scale geographic cells, and train a deep network using millions of geotagged images. We show that the resulting model, called *PlaNet*, outperforms previous approaches and even attains superhuman accuracy in some cases. Moreover, we extend our model to photo albums by combining it with a long short-term memory (LSTM) architecture. By learning to exploit temporal coherence to geolocate uncertain photos, this model achieves a 50 % performance improvement over the single-image model.

1 Introduction

Photo geolocation is an extremely challenging task since many photos offer only few, possibly ambiguous, cues about their location. For instance, an image of a beach could be taken on many coasts across the world. Even when landmarks are present there can still be ambiguity: a photo of the Rialto Bridge could be taken either at its original location in Venice, Italy, or in Las Vegas which has a replica of the bridge! In the absence of discriminative landmarks, humans can fall back on their world knowledge and use cues like the language of street signs or the driving direction of cars to infer the location of a photo. Traditional computer vision algorithms typically lack this kind of world knowledge, relying on the features provided to them during training. Most previous work has therefore focused on restricted subsets of the problem, like landmark buildings [2,40,59], cities where street view imagery is available [10,29,57], or places with enough

I. Kostrikov and J. Philbin—Work done while at Google.

B. Leibe et al. (Eds.): ECCV 2016, Part VIII, LNCS 9912, pp. 37–55, 2016.
DOI: 10.1007/978-3-319-46484-8_3

photos to build structure-from-motion reconstructions that are used for pose estimation [34,44]. In contrast, our goal is to localize any type of photo taken at any location using just its pixels. Only few other works have addressed this task [19,20].

We treat the task of geolocation as a classification problem and subdivide the surface of the earth into a set of geographical cells which make up the target classes. We then train a convolutional neural network (CNN) [49] using millions of geotagged photos. Given a query photo, our model outputs a discrete probability distribution over the earth, assigning each geographical cell a likelihood that the input photo was taken inside it. The resulting model, which we call *PlaNet*, is capable of localizing a large variety of photos. Besides landmark buildings and street scenes, PlaNet can often predict the location of nature scenes like mountains, waterfalls or beaches, with surprising accuracy. In cases of ambiguity, it will often output a distribution with multiple modes corresponding to plausible locations (Fig. 1). Despite being a much simpler and less resource-intensive approach, PlaNet delivers comparable performance to Im2GPS [19,20] which shares a similar goal. A small-scale experiment shows that PlaNet even reaches superhuman performance at the task of geolocating street view scenes. Moreover, we show that the features learned by PlaNet can be used for image retrieval and achieve state-of-the-art results on the INRIA Holidays dataset [25]. Finally, we show that combining PlaNet with an LSTM approach enables it to use context to predict the locations of ambiguous photos, increasing its accuracy on photo albums by 50 %.

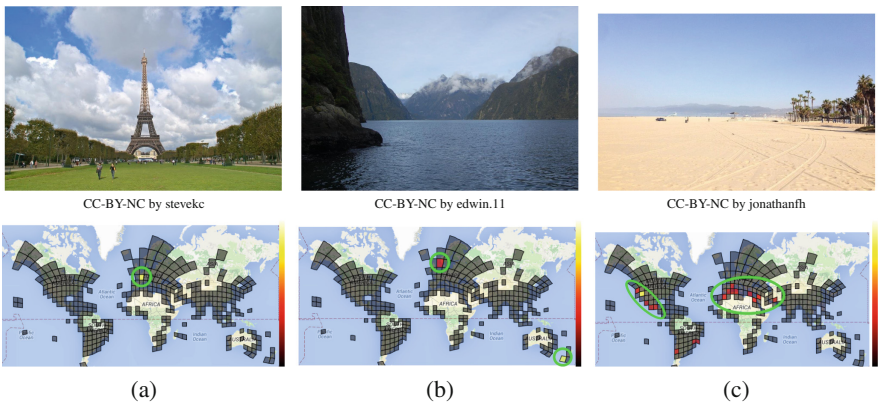


Fig. 1. Given a query photo (top), PlaNet outputs a probability distribution over the surface of the earth (bottom). Viewing the task as a classification problem allows PlaNet to express its uncertainty about a photo. While the Eiffel Tower (a) is confidently assigned to Paris, the model believes that the fjord photo (b) could be taken in either New Zealand or Norway. For the beach photo (c), PlaNet assigns the highest probability to southern California (correct), but some probability is also assigned to places with similar beaches, like Mexico and the Mediterranean. (For visualization purposes we use a model with a much lower spatial resolution than our full model)

2 Related Work

Im2GPS [19,20] (*cf.* Sect. 3) matches a query photo against millions of geotagged Flickr photos using global image descriptors and assigns it the location of the closest match. Because photo coverage in rural areas is sparse, [35,36] make additional use of satellite *aerial imagery*. [36] use CNNs to learn a joint embedding for ground and aerial images and localize a query image by matching it against a database of aerial images. [53] take a similar approach and use a CNN to transform ground-level features to the feature space of aerial images. Local feature based *image retrieval* [38,48] is effective at matching buildings, but requires more space and lacks the invariance to match, *e.g.*, natural scenes or articulated objects. Most local feature based approaches therefore focus on localization within cities, using photos from photo sharing websites [8,39] or street view [4,10,29,30,46,56,57]. Skyline2GPS [41] matches the skyline captured by an upward-facing camera against a 3D model of the city. While matching against geotagged images can provide the rough location of a query photo, *pose estimation* approaches determine the exact 6-dof camera pose of a query image by registering it to a structure-from-motion model [8,23,33,34,43,45]. PoseNet [28] is a CNN that regresses from a query image to its 6-dof pose. However, because a structure-from-motion reconstruction is required for generating its training data, it is restricted to areas with dense enough photo coverage. *Landmark recognition systems* [2,16,24,40,59] construct a database of landmark buildings by clustering internet photo collections and recognize the landmark in a query image using image retrieval. Instead, [7,32] recognize landmarks using SVMs trained on bags-of-visual-words of landmark clusters. [18] perform image geolocation by training one exemplar SVM for each image in a dataset of street view images. CNNs have previously been shown to work well for *scene recognition*. On the SUN database [54], Overfeat [47], a CNN trained on ImageNet [12], consistently outperforms other approaches, including global descriptors like GIST and local descriptors like SIFT, and training on the task-specific Places Database [60] yields another significant boost.

In Sect. 4, we extend PlaNet to geolocate sequences of images using LSTMs. [9,32] address this problem by first clustering the photo collection into landmarks and then learning to predict the sequence of landmarks in a photo sequence. While [9] estimate travel priors on a dataset of photo albums and use a Hidden Markov Model (HMM) to infer the location sequence, [32] train a structured SVM that uses temporal information as an additional feature. [27] also use an HMM, but instead of landmarks, their classes are a set of geographical cells partitioning the surface of the earth, which is similar to our approach. [31] train a CNN on a large collection of geotagged Flickr photos to predict geographical attributes like "population", "elevation" or "household income". In summary, with few exceptions [19,20,35,36,53] most previous approaches to photo geolocation are restricted to urban areas which are densely covered by street view imagery and tourist photos. Prior work has shown that CNNs are well-suited for scene classification [54] and geographical attribute prediction [31], but to our

knowledge ours is the first method that directly takes a classification approach to geolocation using CNNs.

3 Image Geolocation with CNNs

We pose the task of image geolocation as a classification problem and subdivide the earth into a set of geographical cells making up the target classes. The training input to the CNN are the image pixels and the target output is a one-hot vector encoding the cell containing the image. Given a test image, the output of this model is a probability distribution over the world. The advantage of this formulation over a regression from pixels to geo-coordinates is that the model can express its uncertainty about an image by assigning each cell a confidence. A regression model would be forced to pinpoint a single location and would have no natural way of expressing uncertainty, especially in the presence of multi-modal answers (as are expected in this task).

Adaptive Partitioning using S2 Cells. We use Google's open source S2 geometry library[1] to partition the surface of the earth into non-overlapping cells that define the classes of our model. The S2 library defines a hierarchical partitioning of the surface of a sphere by projecting the surfaces of an enclosing cube on it. The six sides of the cube are subdivided hierarchically by six quad-trees. A node in a quad-tree defines a region on the sphere called an S2 cell. Figure 4 illustrates this in 2D. We chose this subdivision scheme over a simple subdivision of lat/lon coordinates, because (i) lat/lon regions get elongated near the poles while S2 cells keep a close-to-quadratic shape, and (ii) S2 cells have mostly uniform size (the ratio between the largest and smallest S2 cell is 2.08).

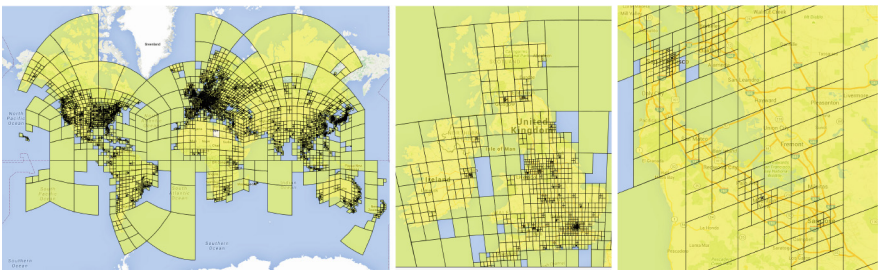


Fig. 2. Left: Adaptive partitioning of the world into 26,263 S2 cells. Middle, Right: Detail views of Great Britain and Ireland and the San Francisco bay area

A naive approach to define a tiling of the earth would use all S2 cells at a certain fixed depth in the hierarchy, resulting in a set of roughly equally sized cells (see Fig. 1). However, this would produce a very imbalanced class distribution since the geographical distribution of photos has strong peaks in densely

[1] https://code.google.com/p/s2-geometry-library/, https://goo.gl/vKikP6.

populated areas. We therefore perform adaptive subdivision based on the photos' geotags: starting at the roots, we recursively descend each quad-tree and subdivide cells until no cell contains more than a certain fixed number t_1 of photos. This way, sparsely populated areas are covered by larger cells and densely populated areas are covered by finer cells. Then, we discard all cells containing less than a minimum of t_2 photos. Therefore, PlaNet does not cover areas where photos are very unlikely to be taken, such as oceans or poles. We remove all images from the training set that are in any of the discarded cells. This adaptive tiling has several advantages over a uniform one: (i) training classes are more balanced, (ii) it makes effective use of the parameter space because more model capacity is spent on densely populated areas, (iii) the model can reach up to street-level accuracy in city areas where cells are small. Figure 2 shows the S2 partitioning for our dataset.

CNN Training. We train a CNN based on the Inception architecture [49] with batch normalization [22]. The SoftMax output layer has one output for each S2 cell in the partitioning. We set the target output to 1.0 for the S2 cell containing the training image and set all others to 0.0. We initialize the model weights with random values and train to minimize the cross-entropy loss using AdaGrad [14] with a learning rate of 0.045.

Our dataset consists of 126M photos with Exif geolocations mined from all over the web. We applied very little filtering, only excluding images that are non-photos (like diagrams, clip-art, etc.) and porn. Our dataset is therefore extremely noisy, including indoor photos, portraits, photos of pets, food, products and other photos not indicative of location. Moreover, the Exif geolocations may be incorrect by several hundred meters due to noise. We split the dataset into 91M training images and 34M validation images.

For the adaptive S2 cell partitioning (Sect. 3) we set $t_1 = 10,000$ and $t_2 = 50$, resulting in 26,263 S2 cells (Fig. 2). Our Inception model has a total of 97,321,048 parameters. We train the model for 2.5 months on 200 CPU cores using the DistBelief framework [11] until the accuracy on the validation set converges. The long training time is due to the large variety of the training data and the large number of classes.

To ensure that none of the test sets we use in this paper have any ll(near-) duplicate images in the training set, we use a CNN trained on near-duplicate images to compute a binary embedding for each training and test image and then remove test images whose Hamming distance to a training image is below an aggressively chosen threshold.

Geolocation Accuracy. We collected a test dataset of 2.3M geotagged Flickr photos from across the world. Other than selecting geotagged images with 1 to 5 textual tags, we did not apply any filtering. Therefore, most of the images have little to no cues about their location. Figure 5 shows example images that illustrate how challenging this benchmark is. We measure localization error as the distance between the center of the predicted S2 cell to the original photo location. We note that this error measure is pessimistic, because even if the ground truth location is within the predicted cell, the error can still be large

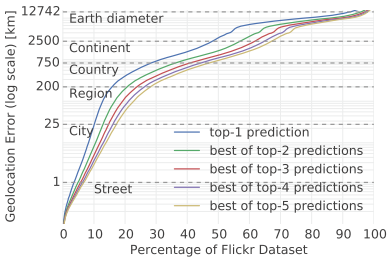


Fig. 3. Geolocation accuracy of the top-k most confident predictions on 2.3M Flickr photos. (Lower right is best) (Color figure online)

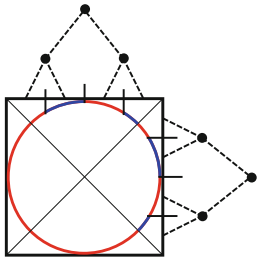


Fig. 4. S2 cell quantization in 2D. The sides of the square are subdivided recursively and projected onto the circle

Fig. 5. Example images drawn randomly from the Flickr test set

depending on the cell size. Figure 3 shows what fraction of this dataset was localized within a certain geographical distance of the ground truth locations. The blue curve shows the performance for the most confident prediction, and the other curves show the performance for the best of the top-{2,3,4,5} predictions per image. Following [20], we added approximate geographical scales of streets, cities, regions, countries and continents. Despite the difficulty of the data, PlaNet is able to localize 3.6 % of the images at street-level accuracy and 10.1 % at city-level accuracy. 28.4 % of the photos are correctly localized at country level and 48.0 % at continent level. When considering the best of the top-5 predictions, the model localizes roughly twice as many images correctly at street, city, region and country level.

Qualitative Results. An important advantage of our localization-as-classification paradigm is that the model output is a probability distribution over the globe. This way, even if an image cannot be confidently localized, the model outputs confidences for possible locations. To illustrate this, we trained a smaller model using only S2 cells at level 4 in the S2 hierarchy, resulting in a total of only 354 S2 cells. Figure 1 shows the predictions of this model for test images with different levels of geographical ambiguity.

Figure 6 shows examples of the different types of images PlaNet can localize. Besides landmarks, which can also be recognized by landmark recognition engines [2,40,59], PlaNet can often correctly localize street scenes, landscapes, buildings of characteristic architecture, locally typical objects like red phone booths, and even some plants and animals. Figure 7 shows some failure modes. Misclassifications often occur due to ambiguity, *e.g.*, because certain landscapes

Fig. 6. Examples of images PlaNet localizes correctly. Our model is capable of localizing photos of famous landmarks (top row), but often yields surprisingly accurate results for images with more subtle geographical cues. The model learns to recognize locally typical landscapes, objects, architectural styles and even plants and animals (Color figure online)

or objects occur in multiple places, or are more typical for a certain place than the one the photo was taken (*e.g.*, the Chevrolet Fleetmaster in the first image is mostly found in Cuba nowadays). To give a visual impression of the representations PlaNet has learned for individual S2 cells, Fig. 8 shows the test images that the model assigns to a given cell with the highest confidence. The model learns a very diverse visual representation of each place, assigning highest confidence to the landmarks, landscapes, or animals that are typical for a specific region.

Comparison to Im2GPS. One of the few approaches that, like ours, aims at geolocating arbitrary photos is Im2GPS [19,20]. However, instead of classification, Im2GPS is based on nearest neighbor matching. The original Im2GPS approach [19] matches the query image against a database of 6.5M Flickr images and returns the geolocation of the closest matching image. Images are represented by a combination of six different global image descriptors. The data was collected by downloading Flickr images that have GPS coordinates and whose

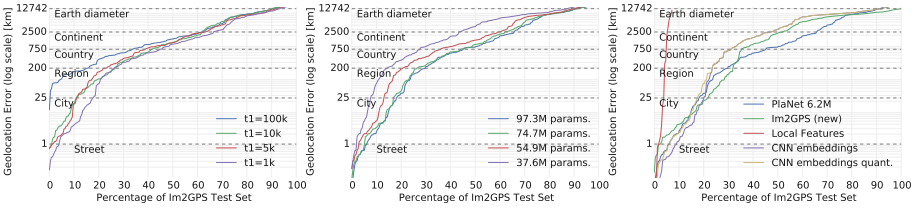
Fig. 7. Examples of incorrectly localized images

Fig. 8. The top-5 most confident images from the Flickr dataset for the S2 cells on the left, showing the diverse visual representation of places that PlaNet learns

tags contain certain geographic keywords including city, country, territory and continent names. To filter out irrelevant content, images tagged with keywords such as "birthday" or "concert" were removed.

A recent extension of Im2GPS [20] uses both an improved image representation and a more sophisticated localization technique. It estimates a per-pixel probability of being "ground", "vertical", "sky", or "porous" and computes color and texture histograms for each of these classes. Additionally, bag-of-visual-word vectors of length 1k and 50k based on SIFT features are computed for each image. The geolocation of a query is estimated by retrieving nearest neighbors, geo-clustering them with mean shift, training 1-vs.-all SVMs for each resulting cluster, and finally picking the average GPS coordinate of the cluster whose SVM gives the query image the highest positive score.

In contrast, PlaNet is a much simpler pipeline. We performed only little filtering to create our input dataset (see above), localization is performed as a straightforward n-way classification, and image features are jointly learned with the classifier parameters during the training process instead of being hand-engineered.



(a) (b) (c)

Fig. 9. (a) Performance for different resolutions of S2 discretization representing different tradeoffs between the number of classes and the number of training images per class (Table 1b). (b) Performance for different numbers of model parameters. The full model has 97.3M parameters. (c) Comparison of PlaNet with image retrieval based on local features and CNN embeddings

Table 1. (a) Comparison of PlaNet with Im2GPS. Percentages are the fraction of images from the Im2GPS test set that were localized within the given radius. (Numbers for the original Im2GPS are approximate as they were extracted from a plot in the paper.) (b) Parameters of PlaNet models with different spatial resolutions

(a)

Method	Street 1 km	City 25 km	Region 200 km	Country 750 km	Continent 2500 km
Im2GPS (orig) [19]		12.0 %	15.0 %	23.0 %	47.0 %
Im2GPS (new) [20]	2.5 %	21.9 %	32.1 %	35.4 %	51.9 %
PlaNet (900k)	0.4 %	3.8 %	7.6 %	21.6 %	43.5 %
PlaNet (6.2M)	6.3 %	18.1 %	30.0 %	45.6 %	65.8 %
PlaNet (91M)	**8.4 %**	**24.5 %**	**37.6 %**	**53.6 %**	**71.3 %**

(b)

t_1	#classes	med. imgs per class	#model
100k	214	21,039	23.9M
10k	2,056	2,140	29.1M
5k	3,852	1,225	34.2M
1k	16,307	320	69.3M

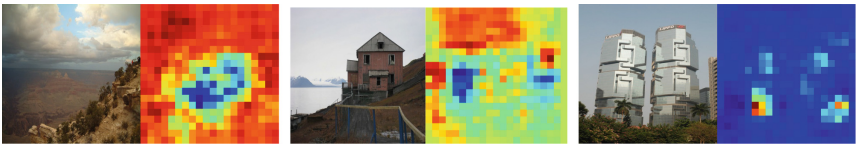
We evaluate PlaNet on the Im2GPS test dataset [19] that consists of 237 geotagged photos from Flickr, curated such that most photos contain at least a few geographical cues. Table 1a compares the performance of three versions of PlaNet trained with different amounts of training data to both versions of Im2GPS. The new Im2GPS version is a significant improvement over the old one. However, the full PlaNet model outperforms even the new version with a considerable margin. In particular, PlaNet localizes 236 % more images accurately at street level. The gap narrows at coarser scales, but even at country level PlaNet still localizes 51 % more images accurately. The 'PlaNet 6.2M' model is more directly comparable to Im2GPS, which uses a database of 6.5M images. While Im2GPS wins on city and region levels, 'PlaNet 6.2M' still outperforms Im2GPS on street, country and continent levels. Using similar amounts of input data, PlaNet shows performance comparable to Im2GPS. However, we note that Im2GPS has an advantage over PlaNet in this experiment, because PlaNet's training set comes from random websites and is thus much more noisy and of lower quality than the Flickr images Im2GPS uses (Flickr is targeted at amateur and professional photographers and thus hosts mainly high quality images). Moreover, because Im2GPS is based on Flickr photos, it has an advantage on this test set which is also mined from Flickr. PlaNet's training data are general web photos which have a different geographical distribution and a higher fraction of irrelevant images with no geographical cues (Fig. 5).

Regardless of accuracy, PlaNet has several advantages over Im2GPS: PlaNet is a single model trained end-to-end, while Im2GPS is a manually engineered pipeline that uses a carefully selected set of features. Furthermore, PlaNet uses much less resources than Im2GPS: Since the Im2GPS feature vectors have a dimensionality of 100,000, Im2GPS would require 8.3 TB to represent our corpus of 91M training examples (577 GB for 6.2M images), assuming one byte per

descriptor dimension. In contrast, PlaNet uses only 377 MB, which even fits into the memory of a smartphone.

Effect of S2 Discretization. An important meta-parameter of PlaNet is the resolution of the S2 cell discretization. A finer discretization means that the number of target classes increases, while the number of training examples per class decreases. At the same time, the number of model parameters increases due to the final fully-connected layer. We trained models with different levels of discretization by varying the t_1 parameter that determines the maximum number of images in a cell, while leaving t_2 fixed at 50 images. We used the same subset of 6.2M training images we used above for comparability with Im2GPS. Table 1b shows the parameters of the different models used and Fig. 9a shows the results. As expected, the lower the resolution, the fewer images can be localized at street accuracy. Interestingly, while the 1k model performs similar to the 5k model at region level and above, it performs significantly better at street level, localizing 4.2 % of the images correctly, while the 5k model only localizes 1.3 % of images at street level. This is surprising since this model has the highest number of parameters and the lowest number of training images per class, making it prone to overfitting. However, it still performs well since the images it can localize at street level are from dense city centers with a fine S2 discretization, where sufficient training examples exist.

Effect of Model Size. To analyze how many model parameters are required, we trained models with reduced numbers of parameters. As Fig. 9b shows, a model with 74.7M parameters performs almost as well as the full model which has 97.3M parameters, but when reducing the number of parameters further, performance starts to degrade.



(a) (b) (c)

Fig. 10. Left: Input image, right: Heatmap of the probability of the correct class when sliding an occluding window over the image [58]. (a) Grand Canyon - Occluding the region containing the distinctive mountain formation makes the confidence in the correct location drop the most. (b) Norway - The snowy mountain range on the left is the most important cue. (c) Hong Kong - Confidence in the correct location *increases* if the palm trees in the foreground are covered since they are not typical for Hong Kong

Comparison to Image Retrieval. We compare the PlaNet 6.2M model with two image retrieval baselines that assign each query photo the location of the closest matching image from the PlaNet 6.2M training data (Fig. 9c). *'Local Features'* retrieves images using an inverted index and spatially verifies tentative

matches [39,48]. *'CNN embeddings'* represents each image as a 256 byte vector extracted with a CNN trained on a landmark dataset using the triplet loss [52]. Matching is performed w.r.t. the L_2 distance. 'Local Features' work well on rigid objects like landmarks, but fail to match, similar scenes, causing its low recall. 'CNN embeddings' outperform PlaNet at street level (9.28 % vs. 6.33 %), but fall behind at region level (23.63 % vs. 29.96 %) and above. PlaNet's disadvantage on street level is that its geolocations are quantized into cells, while the retrieval model can use the exact geolocations. Using the same quantization for retrieval (*'CNN embeddings quant.'*) has the same performance as PlaNet on street and city level. This suggests that both retrieval and classification are well-suited for recognizing specific locations inside cities, but classification seems more suitable for recognizing generic scenes and subtle location cues. We also note that the embeddings use 1.5 GB for 6.2M images and would use 21.7 GB for the full 91M images while the PlaNet model uses only 377 MB regardless of the amount of training data.

Model Analysis. To analyze which parts of the input image are most important for the classifier's decision, we employ a method introduced by Zeiler *et al.* [58]. We plot an activation map where the value of each pixel is the classifier's confidence in the ground truth geolocation if the corresponding part of the image is occluded by a gray box (Fig. 10). The first two examples show that the image regions that would be most useful for a human are also most important for the decision of the model. However, as the last example shows, the model can also be fooled by misleading cues.

Comparison to Human Performance. To find out how PlaNet compares with human intuition, we let it compete against 10 well-traveled human subjects in a game of Geoguessr (www.geoguessr.com). Geoguessr presents the player with a random street view panorama (sampled from all street view panoramas across the world) and asks them to place a marker on a map at the panorama' location. We used the game's "challenge mode" where two players are shown the same set of 5 panoramas. We entered the PlaNet guesses manually by running inference on a screenshot of the view presented by the game and entering the center of the highest confidence S2 cell as its guess. We did not allow the human players to pan, zoom or navigate, so they did not use more information than the model. For each player we used a different set of panoramas, so humans and PlaNet played a total of 50 different rounds. PlaNet won 28 of the 50 rounds with a median localization error of 1131.7 km, while the median human localization error was 2320.75 km. Neither humans nor PlaNet were able to localize photos below street or city level, showing that this task was even harder than the Flickr dataset and the Im2GPS dataset. PlaNet was able to localize twice as many photos at region level (4 vs. 2), 1.54× as many photos at country level (17 vs. 11), and 1.23× as many photos at continent level (32 vs. 26). Figure 11 shows example panoramas with the guessed locations. Most panoramas were taken in rural areas containing little to no geographical cues.

When asked what cues they used, human subjects said they looked for any type of signs, the types of vegetation, the architectural style, the color of lane

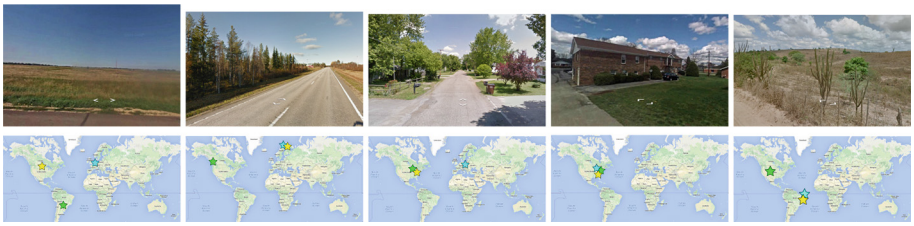


Fig. 11. Top: GeoGuessr panorama, Bottom: Ground truth location (yellow), human guess (green), PlaNet guess (blue) (Color figure online)

markings and the direction of traffic on the street. Furthermore, humans knew that street view is not available in certain countries such as China allowing them to further narrow down their guesses. One would expect that these cues, especially street signs, together with world knowledge and common sense should give humans an unfair advantage over PlaNet, which was trained solely on images and geolocations. Yet, PlaNet was able to outperform humans by a considerable margin. For example, PlaNet localized 17 panoramas at country granularity (750 km) while humans only localized 11 panoramas within this radius. We think PlaNet has an advantage over humans because it has seen many more places than any human can ever visit and has learned subtle cues of different scenes that are even hard for a well-traveled human to distinguish.

Features for Image Retrieval. A recent study [42] showed that the activations of Overfeat [47], a CNN trained on ImageNet [12] can serve as powerful features for several computer vision tasks, including image retrieval. Since PlaNet was trained for location recognition, its features should be particularly suited for image retrieval of touristic photos. To test this, we evaluate the PlaNet features on the INRIA Holidays dataset [25], consisting of 1,491 personal holiday photos, including landmarks, cities and natural scenes and the and Oxford5k dataset [39], consisting of 5,062 images of historic buildings in Oxford. We extract image embeddings from the final layer below the SoftMax layer (a 2048-dim. vector) and rank images by the L_2 distance between their embedding vectors. As can be seen in Table 2a, the PlaNet features outperform the Overfeat features. Using the *spatial search* and *augmentation* techniques described in [42], PlaNet even outperforms state-of-the-art local feature based image retrieval approaches on the Holidays dataset. PlaNet is not as competitive on Oxford since the query images of this dataset are small cut-out image regions, requiring highly scale-invariant matching, which gives local feature based approaches an advantage. We note that the Euclidean distance between these image embeddings is not necessarily meaningful as PlaNet was trained for classification. We expect Euclidean embeddings trained for image retrieval using a triplet loss [52] to deliver even higher mAP.

Table 2. (a) Image retrieval mAP using PlaNet features compared to other methods. (b) Results of PlaNet LSTM on Google+ photo albums. Percentages are the fraction of images in the dataset localized within the respective distance

(a)

Method	Holidays	Oxford
BoVW	57.2 [26]	38.4 [26]
Hamming Embedding	77.5 [26]	56.1 [26]
Fine Vocabulary	74.9 [37]	74.2 [37]
ASMK+MA	82.2 [51]	**81.7** [51]
GIST	37.6 [13]	
Overfeat	64.2 [42]	32.2 [42]
Overfeat+aug+ss	84.3 [42]	68.0 [42]
AlexNet+LM Retraining	79.3 [6]	54.5 [6]
CNN+aug+ss	**90.0** [3]	79.0 [3]
Aggr. local CNN features	80.2 [5]	58.9 [5]
Pooled CNN features+QE		66.9 [50]
NetVLAD	83.1 [1]	71.6 [1]
NBNN on CNN features	88.7 [55]	
PlaNet (this work)	73.3	34.9
PlaNet+aug+ss	89.9	

(b)

Method	Street 1 km	City 25 km	Region 200 km	Country 750 km	Continent 2500 km
PlaNet	14.9 %	20.3 %	27.4 %	42.0 %	61.8 %
PlaNet avg	22.2 %	35.6 %	51.4 %	68.6 %	82.7 %
PlaNet HMM	23.3 %	34.3 %	47.1 %	63.2 %	79.5 %
LSTM	32.0 %	42.1 %	57.9 %	75.5 %	87.9 %
LSTM off1	30.9 %	41.0 %	56.9 %	74.5 %	85.4 %
LSTM off2	29.9 %	40.0 %	55.8 %	73.4 %	85.9 %
LSTM rep	**34.5 %**	**45.6 %**	**62.6 %**	**79.3 %**	**90.5 %**
LSTM rep 25	28.3 %	37.5 %	49.9 %	68.9 %	82.0 %
BLSTM 25	**33.0 %**	**43.0 %**	**56.7 %**	**73.2 %**	**86.1 %**

4 Sequence Geolocation with LSTMs

While PlaNet is capable of localizing a large variety of images, many images are ambiguous or do not contain enough information that would allow to localize them. However we can exploit the fact that photos naturally occur in sequences, *e.g.*, photo albums, with a high geographical correlation. Intuitively, if we can confidently localize some of the photos in an album, we can use this information to also localize the photos with uncertain location. Assigning each photo in an album a location is a sequence-to-sequence problem which requires a model that accumulates a state from previously seen examples and makes the decision for the current example based on both the state and the current example. Therefore, long-short term memory (LSTM) architectures [21] seem like a good fit for this task. Moreover, using LSTMs allows us to express the entire pipeline as a

single neural network. While previous works [9,27] have used HMMs, our results indicate that LSTMs are better suited for this problem.

Training Data and Model Architecture. We collected a dataset of 29.7M public photo albums with geotags from Google+, which we split into 23.5M training albums (490M images) and 6.2M testing albums (126M) images. We use the S2 quantization scheme from the previous section. The basic structure of our model is as follows (Fig. 12a): Given an image, we extract an embedding vector from the final layer before the SoftMax layer in PlaNet. This vector is fed into the LSTM unit. The output vector of the LSTM is then fed into a SoftMax layer that performs the classification into S2 cells. We feed the images of an album into the model in chronological order. For the Inception part, we re-use the parameters of the single-image model. During training, we keep the Inception part fixed and only train the LSTM units and the SoftMax layer.

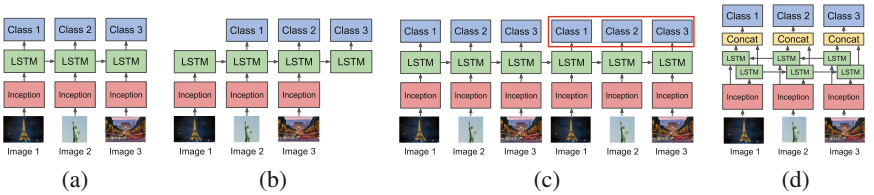


Fig. 12. Time-unrolled diagrams of the PlaNet LSTM models. (a) Basic model. (b) Label offset. (c) Repeated sequence. The first pass is used to generate the state inside the LSTM, so we only use the predictions of the second pass (red box). (d) Bi-directional LSTM (Color figure online)

Results. We compare our LSTM results to three baselines: *'PlaNet'* is our single-image model, *'PlaNet avg'* assigns each image in the album the average of the confidences of the single-image model, *'PlaNet HMM'* is a Hidden Markov Model on top of PlaNet. Like [9,27], we estimate the HMM class priors and transition probabilities by counting and compute the emission probabilities by applying Bayes' rule to the class posterior probabilities of PlaNet. Unlike [27], we do not incorporate time or distance into the transition probability, but use a much finer spatial resolution (26,263 bins vs. 3,186 bins). We determine the maximum likelihood state sequence using the Viterbi algorithm.

The results are shown in Table 2b. 'PlaNet avg' already yields a significant improvement over single-image PlaNet (49.0 % relative on street level), since it transfers more confident predictions to ambiguous images. Interestingly, the HMM does not perform much better than the simple averaging approach. This is surprising since 'PlaNet avg' predicts the same location for all images. However, its advantage over HMMs is that it sees the whole sequence, while the HMM only uses the images before the current one.

The LSTM model clearly outperforms the averaging and HMM (44.1 % and 37.3 % relative improvement over single-image on the street level, respectively).

Visual inspection of results showed that if an image with high location confidence is followed by several images with lower location confidence, the LSTM model assigns the low-confidence images locations close to the high-confidence image. Thus, while the original PlaNet model tends to "jump around", the LSTM model tends to predict close-by locations unless there is strong evidence of a location change. The LSTM model outperforms the averaging baseline because averaging assigns all images in an album the same confidences and can thus not produce accurate predictions for albums that include different locations (such as albums of trips). The LSTM model outperforms the HMM model, because it is able to capture long-term relationships. For example, at a given photo, HMMs will assign high transition probabilities to all neighboring locations due to the Markov assumption, while LSTMs are capable of learning specific tourist routes conditioned on previous locations. A problem with this simple LSTM model is that many albums contain a number of images in the beginning that contain no helpful visual information. Due to its unidirectional nature, this model cannot fix wrong predictions that occur in the beginning of the sequence after observing a photo with a confident location. For this reason, we now evaluate a model where the LSTM ingests multiple photos from the album before making its first prediction.

Label Offset. The idea of this model is to shift the labels such that inference is postponed for several time steps (Fig. 12b) The main motivation under this idea is that this model can accumulate information from several images in a sequence before making predictions. Nevertheless, we found that using offsets does not improve localization accuracy (Table 2b, LSTM off1, LSTM off2). We assume this is because the mapping from input image to output labels becomes more complex, making prediction more difficult for all photos, while improving predictions just for a limited amount of photos. Moreover, this approach does not solve the problem universally: For example, if we offset the label by 2 steps, but the first image with high location confidence occurs only after 3 steps, the prediction for the first image will likely still be wrong. To fix this, we now consider models that condition their predictions on all images in the sequence instead of only previous ones.

Repeated Sequences. We first evaluate a model that was trained on sequences that had been constructed by concatenating two instances of the same sequence (Fig. 12c) For this model, we take predictions only for the images from the second half of the sequence (*i.e.* the repeated part). Thus, all predictions are conditioned on observations from all images. At inference time, passing the sequence to the model for the first time can be viewed as an *encoding* stage where the LSTM builds up an internal state based on the images. The second pass is the *decoding* stage where the LSTM makes predictions based on its state and the current image. Results show that this approach outperforms the single-pass LSTMs (Table 2b, 'LSTM rep'), achieving a 7.8 % relative improvement at street level, at the cost of a twofold increase in inference time. However, visual inspection showed a problem with this approach: if there are low-confidence images at the beginning of the sequence, they tend to get assigned to the last confident

location in the sequence, because the model learns to rely on its previous prediction. Therefore, predictions from the end of the sequence get carried over to the beginning.

Bi-directional LSTM. A well-known neural network architecture that conditions the predictions on the whole sequence are bi-directional LSTM (BLSTM) [17]. This model can be seen as a concatenation of two LSTM models, where the first one does a forward pass, while the second does a backward pass on a sequence (Fig. 12d). Bi-directional LSTMs cannot be trained with truncated back-propagation through time [15] and thus require to unroll the LSTMs to the full length of the sequence. To reduce the computational cost of training, we had to limit the length of the sequences to 25 images. This causes a decrease in total accuracy since longer albums typically yield higher accuracy than shorter ones. Since our experiments on this data are not directly comparable to the previous ones, we also evaluate the repeated LSTM model on sequences truncated to 25 images. As the results show (Table 2b: 'LSTM rep 25', 'BLSTM 25'), BLSTMs clearly outperform repeated LSTMs (16.6 % relative improvement on street level). However, because they are not tractable for long sequences, the repeated model might still be preferable in practice.

5 Conclusion

We presented PlaNet, a CNN for image geolocation. Regarding the problem as one of classification, PlaNet produces a probability distribution over the globe. This allows it to express its uncertainty about the location of a photo and assign probability mass to potential locations. While previous work mainly focused on photos taken inside cities, PlaNet is able to localize landscapes, locally typical objects, and even plants and animals. Our experiments show that PlaNet far outperforms other methods for geolocation of generic photos and even reaches superhuman performance. We further extended PlaNet to photo album geolocation by combining it with LSTMs, achieving 50 % higher performance than the single-image model.

References

1. Arandjelovic, R., Gronat, P., Torii, A., Pajdla, T., Sivic, J.: NetVLAD: CNN architecture for weakly supervised place recognition. In: CVPR (2016)
2. Avrithis, Y., Kalantidis, Y., Tolias, G., Spyrou, E.: Retrieving landmark and non-landmark images from community photo collections. In: ACM Multimedia, pp. 153–162 (2010)
3. Azizpour, H., Razavian, A.S., Sullivan, J., Maki, A., Carlsson, S.: From generic to specific deep representations for visual recognition. In: CVPR DeepVision Workshop (2015)
4. Baatz, G., Köser, K., Chen, D., Grzeszczuk, R., Pollefeys, M.: Handling urban location recognition as a 2D homothetic problem. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part VI. LNCS, vol. 6316, pp. 266–279. Springer, Heidelberg (2010)

5. Babenko, A., Lempitsky, V.: Aggregating local deep features for image retrieval. In: ICCV (2015)
6. Babenko, A., Slesarev, A., Chigorin, A., Lempitsky, V.: Neural codes for image retrieval. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014, Part I. LNCS, vol. 8689, pp. 584–599. Springer, Heidelberg (2014)
7. Bergamo, A., Sinha, S.N., Torresani, L.: Leveraging structure from motion to learn discriminative codebooks for scalable landmark classification. In: CVPR, pp. 763–770 (2013)
8. Cao, S., Snavely, N.: Graph-based discriminative learning for location recognition. IJCV **112**(2), 239–254 (2015)
9. Chen, C.Y., Grauman, K.: Clues from the beaten path: location estimation with bursty sequences of tourist photos. In: CVPR (2011)
10. Chen, D., Baatz, G., Köser, K., Tsai, S., Vedantham, R., Pylvänäinen, T., Roimela, K., Chen, X., Bach, J., Pollefeys, M., Girod, B., Grzeszczuk, R.: City-scale landmark identification on mobile devices. In: CVPR, pp. 737–744 (2011)
11. Dean, J., Corrado, G.S., Monga, R., Chen, K., Devin, M., Le, Q.V., Mao, M.Z., Ranzato, M., Senior, A., Tucker, P., Yang, K., Ng, A.Y.: Large scale distributed deep networks. In: NIPS (2012)
12. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNet: a large-scale hierarchical image database. In: CVPR (2009)
13. Douze, M., Jégou, H., Harsimrat, S., Amsaleg, L., Schmid, C.: Evaluation of GIST descriptors for web-scale image search. In: CIVR (2009)
14. Duchi, J., Hazan, E., Singer, Y.: Adaptive subgradient methods for online learning and stochastic optimization. JMLR **12**, 2121–2159 (2011)
15. Elman, J.: Finding structure in time. Cogn. Sci. **14**(2), 179–211 (1990)
16. Gammeter, S., Quack, T., Van Gool, L.: I know what you did last summer: object-level auto-annotation of holiday snaps. In: ICCV, pp. 614–621 (2009)
17. Graves, A., Schmidthuber, J.: Framewise phoneme classification with bidirectional LSTM and other neural network architectures. Neural Netw. **18**(5–6), 602–610 (2005)
18. Gronat, P., Obozinski, G., Sivic, J., Pajdla, T.: Learning per-location classifiers for visual place recognition. In: CVPR (2013)
19. Hays, J., Efros, A.: IM2GPS: estimating geographic information from a single image. In: CVPR (2008)
20. Hays, J., Efros, A.: Large-scale image geolocalization. In: Choi, J., Friedland, G. (eds.) Multimodal Location Estimation of Videos and Images, pp. 41–62. Springer, Cham (2014)
21. Hochreiter, S., Schmidthuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
22. Ioffe, S., Szegedy, C.: Batch normalization: accelerating deep network training by reducing internal covariate shift. In: ICML (2015)
23. Irschara, A., Zach, C., Frahm, J.M., Bischof, H.: From structure-from-motion point clouds to fast location recognition. In: CVPR (2009)
24. Johns, E., Yang, G.Z.: From images to scenes: compressing an image cluster into a single scene model for place recognition. In: ICCV, pp. 874–881 (2011)
25. Jegou, H., Douze, M., Schmid, C.: Hamming embedding and weak geometric consistency for large scale image search. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part I. LNCS, vol. 5302, pp. 304–317. Springer, Heidelberg (2008)
26. Jégou, H., Douze, M., Schmid, C.: Improving bag-of-features for large scale image search. IJCV **87**(3), 316–336 (2010)

27. Kalogerakis, E., Vesselova, O., Hays, J., Efros, A., Hertzmann, A.: Image sequence geolocation with human travel priors. In: ICCV (2009)
28. Kendall, A., Grimes, M., Cipolla, R.: PoseNet: a convolutional network for real-time 6-DOF camera relocalization. In: ICCV (2015)
29. Kim, H.J., Dunn, E., Frahm, J.M.: Predicting good features for image geo-localization using per-bundle VLAD. In: ICCV (2015)
30. Knopp, J., Sivic, J., Pajdla, T.: Avoiding confusing features in place recognition. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part I. LNCS, vol. 6311, pp. 748–761. Springer, Heidelberg (2010)
31. Lee, S., Zhang, H., Crandall, D.J.: Predicting geo-informative attributes in large-scale image collections using convolutional neural networks. In: WACV (2015)
32. Li, Y., Crandall, D.J., Huttenlocher, D.P.: Landmark classification in large-scale image collections. In: ICCV, pp. 1957–1964 (2009)
33. Li, Y., Snavely, N., Huttenlocher, D.P.: Location recognition using prioritized feature matching. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part II. LNCS, vol. 6312, pp. 791–804. Springer, Heidelberg (2010)
34. Li, Y., Snavely, N., Huttenlocher, D., Fua, P.: Worldwide pose estimation using 3D point clouds. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012, Part I. LNCS, vol. 7572, pp. 15–29. Springer, Heidelberg (2012)
35. Lin, T.Y., Belongie, S., Hays, J.: Cross-view image geolocalization. In: CVPR (2013)
36. Lin, T.Y., Cui, Y., Belongie, S., Hays, J.: Learning deep representations for ground-to-aerial geolocalization. In: CVPR (2015)
37. Mikulík, A., Perdoch, M., Chum, O., Matas, J.: Learning a fine vocabulary. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010. LNCS, vol. 6313, pp. 1–14. Springer, Heidelberg (2010). doi:10.1007/978-3-642-15558-1_1
38. Nistér, D., Stewénius, H.: Scalable recognition with a vocabulary tree. In: CVPR, pp. 2161–2168 (2006)
39. Philbin, J., Chum, O., Isard, M., Sivic, J., Zisserman, A.: Object retrieval with large vocabularies and fast spatial matching. In: CVPR (2007)
40. Quack, T., Leibe, B., Van Gool, L.: World-scale mining of objects and events from community photo collections. In: CIVR, pp. 47–56 (2008)
41. Ramalingam, S., Bouaziz, S., Sturm, P., Brand, M.: SKYLINE2GPS: localization in urban canyons using omni-skylines. In: IROS (2010)
42. Razavian, A.S., Azizpour, H., Sullivan, J., Carlsson, S.: CNN features off-the-shelf: an astounding baseline for recognition. In: CVPR 2014 DeepVision Workshop (2014)
43. Sattler, T., Leibe, B., Kobbelt, L.: Fast image-based localization using direct 2d-to-3d matching. In: ICCV, pp. 667–674 (2011)
44. Sattler, T., Leibe, B., Kobbelt, L.: Improving image-based localization by active correspondence search. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012, Part I. LNCS, vol. 7572, pp. 752–765. Springer, Heidelberg (2012)
45. Sattler, T., Weyand, T., Leibe, B., Kobbelt, L.: Image retrieval for image-based localization revisited. In: BMVC, pp. 76.1–76.12 (2012)
46. Schindler, G., Brown, M., Szeliski, R.: City-scale location recognition. In: CVPR (2007)
47. Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., LeCun, Y.: OverFeat: integrated recognition, localization and detection using convolutional networks. In: ICLR (2014)

48. Sivic, J., Zisserman, A.: Video Google: a text retrieval approach to object matching in videos. In: ICCV, vol. 2, pp. 1470–1477 (2003)
49. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: CVPR (2015)
50. Tolias, G., Sicre, R., Jegou, H.: Particular object retrieval with integral max-pooling of CNN activations. In: ICLR (2016)
51. Tolias, G., Avrithis, Y., Jegou, H.: To aggregate or not to aggregate: selective matchkernels for image search. In: ICCV (2013)
52. Wang, J., Song, Y., Leung, T., Rosenberg, C., Wang, J., Philbin, J., Chen, B., Wu, Y.: Learning fine-grained image similarity with deep ranking. In: CVPR (2014)
53. Workman, S., Souvenir, R., Jacobs, N.: Wide-area image geolocalization with aerial reference imagery. In: ICCV (2015)
54. Xiao, J., Ehinger, K.A., Hays, J., Torralba, A., Oliva, A.: SUN database: exploring a large collection of scene categories. IJCV (2014)
55. Xie, L., Hong, R., Zhang, B., Tian, Q.: Image classification and retrieval are ONE. In: ICMR (2015)
56. Zamir, A.R., Shah, M.: Accurate image localization based on Google maps street view. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part IV. LNCS, vol. 6314, pp. 255–268. Springer, Heidelberg (2010)
57. Zamir, A.R., Shah, M.: Image geo-localization based on multiple nearest neighbor feature matching using generalized graphs. PAMI **36**(8), 1546–1558 (2014)
58. Zeiler, M.D., Fergus, R.: Visualizing and understanding convolutional networks. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014, Part I. LNCS, vol. 8689, pp. 818–833. Springer, Heidelberg (2014)
59. Zheng, Y.T., Zhao, M., Song, Y., Adam, H., Buddemeier, U., Bissacco, A., Brucher, F., Chua, T.S., Neven, H.: Tour the world: building a web-scale landmark recognition engine. In: CVPR, pp. 961–962 (2009)
60. Zhou, B., Lapedriza, A., Xiao, J., Torralba, A., Oliva, A.: Learning deep features for scene recognition using places database. In: NIPS (2014)