Patch PlaNet: Landmark Recognition with Patch Classification Using Convolutional Neural Networks

Kelvin B. da Cunha, Lucas Maggi, Veronica Teichrieb [*]
[*] Voxar Labs - Centro de Informática
Universidade Federal de Pernambuco
Recife, Brazil
{kbc, lom, vt}@cin.ufpe.br

João Paulo Lima [†*]
[†] Departamento de Computação
Universidade Federal Rural de Pernambuco
Recife, Brazil
joao.mlima@ufrpe.br

Jonysberg Peixoto Quintino [‡]
[‡] Projeto de P&D CIn/Samsung
Universidade Federal de Pernambuco
Recife, Brazil
jpq@cin.ufpe.br

Fabio Q. B. da Silva, Andre L M Santos [§]
[§] Centro de Informática
Universidade Federal de Pernambuco
Recife, Brazil
{fabio, alms}@cin.ufpe.br

Helder Pinho [¶]
[¶] Samsung Instituto de
Desenvolvimento para a Informática
Campinas, Brazil
helder.p@sidi.org.br

***Abstract*—In this work we address the problem of landmark recognition. We extend PlaNet, a model based on deep neural networks that approaches the problem of landmark recognition as a classification problem and performs the recognition of places around the world. We propose an extension of the PlaNet technique in which we use a voting scheme to perform the classification, dividing the image into previously defined regions and inferring the landmark based on these regions. The prediction of the model depends not only on the information of the features learned by the deep convolutional neural network architecture during training, but also uses local information from each region in the image for which the classification is made. To validate our proposal, we performed the training of the original PlaNet model and our variation using a database built with images from Flickr, and evaluated the models in the Paris and Oxford Buildings datasets. It was possible to notice that the addition of image division and voting structure improves the accuracy result of the model by 5-11 percentage points on average, reducing the level of ambiguity found during the inference of the model.**

I. Introduction

Performing the task of landmark recognition using only the information contained in the pixels of an image is a very challenging task.

One of the main difficulties for recognition algorithms is the wide variety of landmarks existing around the world. This large amount of information increases the difficulty for the algorithm to learn relevant and distinct characteristics between classes in order to allow good classification results. Solving this problem efficiently can affect the complexity and the runtime performance of the model used.

Another great challenge in the task of recognizing landmarks is the ambiguity existing between the characteristics and the visual aspect of some constructions and regions. For example, there are similar architectural styles among landmarks from the same country or the same era when they were built. It can also be considered the cases of natural structures of certain environments, which independent of their locality will always have similar characteristics, as shown in Figure 1.

Human beings are able to overcome these challenges by using prior knowledge of the main characteristics among landmarks, such as architectural styles used in a specific region or country, materials used by certain types of buildings and so on. When it is not possible to locate the landmark with certainty using only the pixels information, the human being is able to estimate an approximate location for the landmark, giving a list of probable places to which the landmark may belong.

A recognition algorithm to solve the landmark recognition task must perform a prediction in order to solve these challenges. Some techniques propose ways to generate a model for learning the main characteristics as good as humans in order to correctly classify existing landmarks [1], [2].

The challenge for computational models is to generate such characteristics so that they are sufficiently relevant to distinguish well all known landmarks [2]–[4]. Some techniques use the intuition of the human strategy for recognition, developing an algorithm capable of returning a distribution of probabilities among the most likely regions to which the input image may belong, improving its classification performance [5].

Recent studies address these problems of features design using deep neural network models [6]–[8] to automatically learn the best set of characteristics and the relation between existing landmarks for classification and to return a set of solutions that correspond to the probability distribution of the most likely places where a given input image was captured.

In this work we use an approach based on [7] that uses deep convolutional neural networks (DCNNs) to perform the landmark recognition task by solving a classification problem.

We perform PlaNet model replication (training with our dataset) and validation in widely used datasets [9], [10] that contains locations from Paris and Oxford. We use 7 classes from Paris dataset and 11 classes from the Oxford dataset,

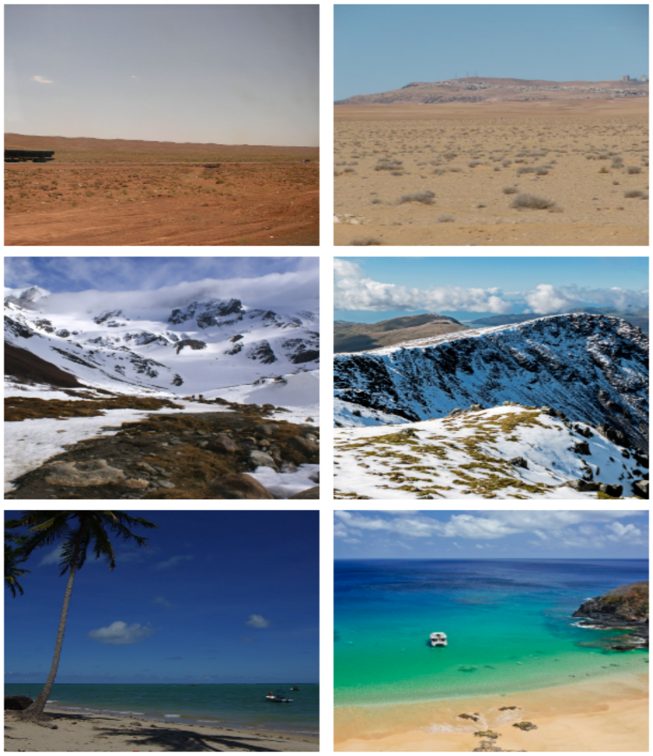

Fig. 1. Comparison between natural landmarks that have similar characteristics but belong to different localities. The top left image corresponds to the desert between Tehran and Isfahan (Reprinted from Flickr by Jordan Lundqvist, 2006. Retrieved from https://flic.kr/p/fkV7d), while the top right one corresponds to a desert in a Jordan highway (Reprinted from Flickr, by Steven Damron, 2010. Retrieved from https://flic.kr/p/7vuZfZ). In the middle the left image shows the Ushuaia mountain (Reprinted from Flickr, by Arturo A. Galn, 2011. Retrieved from https://flic.kr/p/azSjW7) and the right one depicts the Cader Idris mountain (Reprinted from Flickr, by Robert J. Heath, 2016. Retrieved from https://flic.kr/p/WtEs9j). In bottom left we can see a photo taken on the Maragogi beach (Reprinted from Flickr, by Guilherme Jofili, 2009. Retrieved from https://flic.kr/p/7i7Maa) and in bottom right a photo of the Fernando de Noronha island (Reprinted from Flickr, by Leandro M. Gonçalves, 2012. Retrieved from https://flic.kr/p/hyJPA1).

training our model with 18 classes. As a contribution, we provide a technique for image division and voting that improves the accuracy of the original model. We trained our variation in the same conditions as the PlaNet model and evaluated with the same dataset. It is possible to note that this variation is able to increasing the model perfomance by 4.8 percentage points.

A major challenge for DCNN-based landmark recognition algorithms is the availability of data for training. In [7] millions of images from places around the world were used for model training during several months. Such images are not publicly available.

The dataset used directly influences the performance that a model can have, depending on the amount of data and possible variations of each class to learn a good representation for solving the task.

For this reason, we collected and analyzed a new set of images to enable the training of our PlaNet implementation.

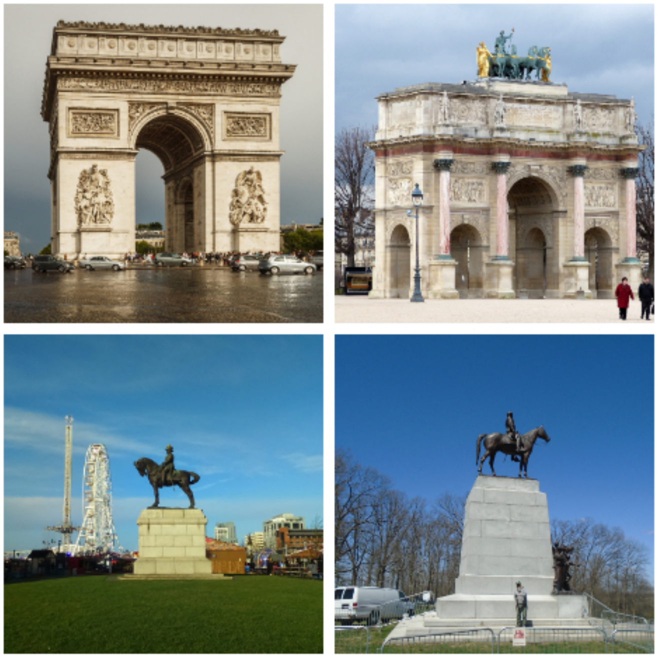

Fig. 2. Comparison between buildings that have similar characteristics but belong to different localities. In these examples, humans can easily differentiate the landmarks by observing features in small regions of the construction or the surrounding environment. The difficulty for computational vision algorithms is to capture this ability to distinguish the target from secondary features. The top left image corresponds to the the Arc de Triomphe de l'toile (Reprinted from Flickr by Greg Men, 2014. Retrieved from https://flic.kr/p/H7b1SE) that has an architecture similar to the top right image from the Arc de Triomphe du Carrousel (Reprinted from Flickr, by Chris Dart, 2013. Retrieved from https://flic.kr/p/e6NE9h). In bottom left we can see a photo taken of the King Edward VII statue at Liverpool Pier Head (Reprinted from Flickr, by Ninian Reid, 2017. Retrieved from https://flic.kr/p/21GHcHi) and in bottom right a photo of the Robert E. Lee statue at Gettysburg island (Reprinted from Flickr, by Meghan, 2014. Retrieved from https://flic.kr/p/26imVKw).

Due to the large difference between the dataset scales, we apply a data augmentation procedure, performing a series of transformations to the obtained images. With this, we approximate the data volume used to train and improve our model performance compared to the original model.

Note that we can not use the same images and classes used for the original PlaNet training. Due to this, we are validating our approach using a subcase of the landmark recognition problem, targeting some classes beloging to the public datasets [9], [10].

II. Related Work

The task of landmark recognition is a great challenge in computer vision, due to the wide variety of sites around the world, the similarities between characteristics of locations in different regions (such as beaches, mountains, forests, etc.), ambiguities between similar buildings and several variations that the environment can have, such as climatic conditions, changes between day and night, presence of people or objects, etc. This is shown in Figure 2.

Due to the large scale that the problem can cover and techniques limitations, many works are limited to solving parts of the landmark recognition problem, dealing only with cases of urban areas (buildings) [11], [12], natural areas [13], specific regions and cities with Google Street View images available [6] or aerial images [14], [15].

Many works approach the problem of recognizing landmarks with image retrieval methods [3], [4], [11], [16] using handcrafted or CNN features as input for a model that computes the similarities between query and training images.

In recent years deep learning was used to develop most of the currently proposed methods. Methods that use DCNN can outperform traditional techniques, besides allowing algorithms to carry out global geolocalization with good performance. Very few works have addressed the task of localizing any type of photo taken at any location [1], [5], [7], [8].

IM2GPS [1] was the first method to propose global geolocalization. The IM2GPS algorithm employs handcrafted features for scene recognition and uses a nearest neighbor search (NNS) method in a dataset of geotagged images to perform the classification. An extension of IM2GPS [5] was proposed using a DCNN to automatically create the features that will be analyzed for classification. This new method may be used to generate an intermediate representation to be classified using the NNS method or be treated as a solution for a classification problem that generates as response the location in which the image was classified. This method improves the previous IM2GPS result, showing that DCNNs can perform better than handcrafted features in developing filters that extract characteristics relevant to landmark recognition.

PlaNet [7] treats the task of geolocation as a classification problem and subdivides the surface of the earth into a set of geographical cells that make up the target classes. The cells are divided in relation to the number of examples of the region. The greater the variety of images of a certain location, the greater the granularity of the cells in that region. For example, city areas with large numbers of distinct landmarks contain more cells than oceanic areas, where the examples are quite similar.

The model is capable of localizing a large variety of photos without constraints, recognizing cases of nature scenes, mountains, street scenes or beaches with high accuracy. In cases of ambiguity, it will often output a probability distribution ranking the landmark predictions by similarity with the examples.

However, the PlaNet response may have a decrease in performance when the number of classes grows or when the classes in the dataset have a high correlation of characteristics. These cases can cause PlaNet to fail, especially when there is irrelevant information in the input image, or when the data in certain regions of the image resemble the various known classes.

For this reason we propose an extension of PlaNet in which we use DCNN to classify only defined regions within the image and then vote from the predictions of these regions, defining which are the most probable classes for a given input image.

III. Method Overview

The input to our model is an RGB image which is divided into patches of the same size. The patches are processed by the network to output a distribution of probability on all known landmarks. The outputs of each patch are combined into a voting scheme that returns a ranked list of most likely landmarks for the input image. We present each part now in more detail.

A. Landmark Recognition with CNN

The approach employed to recognize landmarks is based on the PlaNet model [7]. PlaNet is built to determine the location where a photo was taken using only pixels information. Differently from most computer vision methods that approach photo geolocation using image retrieval techniques based on geometric features, PlaNet poses the problem as a classification one by subdividing the surface of the earth into thousands of multiscale geographic cells and predicting the photo location across these cells. Then the problem becomes to associate each region with a particular cell and to predict the estimated location of that region in relation to the position on the grid which the cell occupies.

The PlaNet model is composed of a DCNN trained using geotagged images. At inference time, the model output is a discrete probability distribution over the earth, assigning a confidence value to each known cell around the world.

Based on the training examples, the DCNN is able to automatically learn main features to be extracted from the image. At each layer the network can generate a set of associated filters specialized in solving parts of the problem, where low-level characteristics are extracted in the initial layers, increasing the complexity of features in each level of the architecture.

We chose PlaNet as base because it was the first method that directly takes a classification approach to geolocation and can perform prediction on a global scale without image constraints and with good performance.

The architecture employed has less parameters compared to other DCNN architectures. It has a layered structure that allows to simultaneously add responses of spatial information on different scales and does not need fully connected layers to perform prediction.

B. Patches Classification

We use patches from the input image to enhance the classification, an approach named Patch PlaNet. By filtering some parts of the image, we assume that noisy elements could be removed from the decision making for the DCNN. We have used three patterns of patch generation:

- Four equal rectangles: Each rectangle covering exactly one quarter of the image, with a common corner in the middle of the image. (Figure 3.a)
- Five equal rectangles: Same as ”Four equal rectangles”, extended with an equal rectangle in size, but centered on image’s center. (Figure 3.b)

- Six rectangles: Same as ”Five equal rectangles”, extended with a full image rectangle. (Figure 3.c)

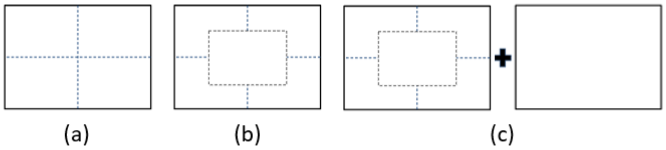

Fig. 3. Patches subdivisions used. Given an image of width w and height h. In (a) we divide the image in four regions with size $\frac{w}{2}$, $\frac{h}{2}$. In (b) we divide the image with the same four regions, adding a new central patch that shares information among the others. In (c) we use the same patches as (b) and add the information of the full image.

C. Classification Methods

We use the patches classification values to reclassify the image, using a voting/ranking method to solve bad guesses (Figure 4). The patches classification was done with three methods for voting/ranking:

- Simple Voting (SV): We assume that all the patches have the same weight in voting, each one having only a single vote, which is the class with higher probability for the patch.
- Weighted Voting (WV): We assume that all the patches have the same weight in voting, but vote with different weights for the N classes with higher probabilities calculated for the patch, according to each class probability result normalized by the sum of the classes probabilities.
- Proportional Weighted Ranking (PWR): We assume that the N classes with higher probabilities are a sample of the overall classes for each patch. Each patch inserts N pairs (probability, class id) in the rank, and all the pairs are then sorted from its probability value.

IV. Implementation Details

The training code was built using the Python programming language and the TensorFlow machine learning framework [17].

We use the Inception V3 [18] architecture with batch normalization [19] and an input size of 299x299x3. We initialize the network using the weights previously trained for the classification task of the ImageNet dataset [20]. In the training phase we employ transfer learning by freezing the parameters for the initial layers and fine-tuning the network in the last layers (Mixed6a to Mixed7c) to learn new filters that extract relevant features for representing our locations.

For the model optimization we use the AdaGrad [21] stochastic gradient descent algorithm with learning rate of 0.045, which is the same as original PlaNet, and set the output cell to be 1.0 when the training sample belongs to it and all others cells to be 0.0. The convolutional layers use ReLU activation while in the last layer the confidence value is computed using Softmax activation.

We apply data augmentation in the training phase to increase the quantity and variety of data and reduce overfitting. We apply the following random image transformations: rotation, vertical and horizontal flip, histogram equalization, grayscale conversion, zoom in and changes in hue, exposure and saturation.

To evaluate the model and improve the inference runtime performance, it was developed a C++ code that loads the network weights trained with Python and performs image classification with the CNN architecture using patches classification.

V. Experiments

In this section, we show the experiments performed during the development of this work. We explain the steps of building our dataset for training and the data used in the evaluation of the technique. We also show the quantitative results obtained for each experiment, and we comment on some cases of robustness and failures that we can observe visually using some qualitative results.

We have trained the models on a computer with an Intel I7-4700MQ @2.4GHz processor and a Nvidia GeForce GTX 950M GPU. The inference was performed on a computer with an Intel Core I5-5200U CPU @2.2GHz processor.

A. Dataset

Our training dataset was built downloading Flickr [22] images with licenses that allow free usage and modification. We got approximately 1,000 samples of each evaluated location and performed data augmentation, due to the low amount of data. After this operation the samples size of each location increased by 10 times. This value was defined based on tests performed with different values.

We validated our implementation using the public datasets with landmark locations of Paris [9] and Oxford buildings [10]. We evaluated 7 landmarks from Paris and 11 landmarks from Oxford, training the model with all these 18 landmarks. These datasets are widely used for validation of image retrieval models [12], [16], [23], [24], [25].

B. Quantitative Results

We compare our results with recent works in image retrieval that use these datasets to evaluate their models. PlaNet was originally trained for global landmark recognition and the original trained model is not publicly available, as well as the dataset used for training.

For this reason, we conducted PlaNet training with our dataset and adapted the model for the recognition of Paris and Oxford Buildings landmarks. With this, we could validate and compare the performance of the model proposed by PlaNet with the other works. The model generated by us achieves a accuracy that is best or comparable to [2], [5], [12] increasing on average 4 percentage points the result for Paris dataset and with a less than 3 percentage points of the result of classification in Oxford dataset.

The results of the validation using Paris and Oxford Buildings datasets can be seen in Table I. Our result using PlaNet for landmark recognition achieved better classification results

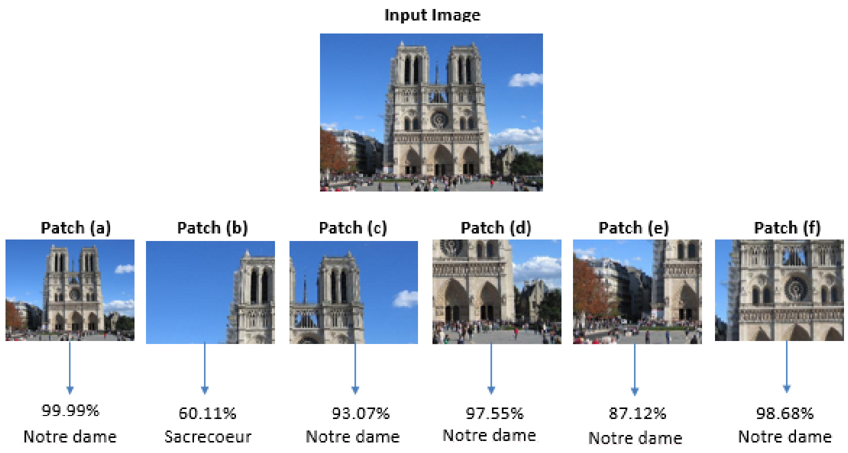

Fig. 4. Example of classification with patches using 6 patches subdivision and PWR. The input image is divided in patches (a), (b), (c), (d), (e) and (f). The PlaNet model will compute a probability distribution for each patch. Then the prediction list is ranked by the highest confidence value. In this example, the most confident answer is shown with its probability for each patch.

than the [2], [5], [12] for the Paris dataset; it is possible to see that using the patch based approach can improve the accuracy of the PlaNet model increasing its accuracy value by 4.8 percentage points.

For the Oxford dataset the performance did not outperform all approaches. This is due to the large number of indoor images and high level of occlusion of the dataset. Our architecture was trained with focus on the recognition of outdoor landmarks, but in the Oxford dataset there are many images where the focus is not the landmark, affecting the performance of our model.

However, it is possible to see that the result obtained for the Oxford dataset remains comparable to the other works.

TABLE I
CLASSIFICATION ACCURACY RESULTS IN PARIS AND OXFORD BUILDINGS DATASETS

Model	Paris Dataset	Oxford Dataset
Patch Planet	**90.63**%	76.03%
PlaNet [7]	85.83%	64.94%
IM2GPS [5]	87.42%	77.30%
Crow [2]	-	74.30%
R-Clean [12]	86.80%	**87.30**%

TABLE II
RESULTS COMBINING THE VARIATIONS IN VOTING SCHEME AND NUMBER OF PATCHES EVALUATED USING THE PARIS DATASET

	Paris dataset		
	SV	WV	PWR
4 Patches	71.72%	79.28%	**82.39**%
5 Patches	75.83%	81.82%	**89.33**%
6 Patches	89.19%	84.14%	**90.63**%

TABLE III
RESULTS COMBINING THE VARIATIONS IN VOTING SCHEME AND NUMBER OF PATCHES EVALUATED USING THE OXFORD DATASET

	Oxford dataset		
	SV	WV	PWR
4 Patches	47.51%	53.45%	**61.65**%
5 Patches	55.86%	59.93%	**69.85**%
6 Patches	63.39%	65.13%	**76.03**%

The accuracy for each class in the Paris and Oxford dataset using PlaNet and PlaNet with patches variations can be seen in Table II and III. The best results can be seen highlighted in the table. Patch PlaNet with 6 patches using PWR obtained the best accuracy results for most of the evaluated landmarks.

With this we can see that the best performance is based on the choice of the region in which the classifier has the greatest confidence that the landmark is present. Predicting directly with the information of the full input image when it has a high confidence value and using information from the patches when it is not so sure which landmark is the correct one.

In Table IV and V, it is possible to see the accuracy for each class in the Paris dataset and Oxford dataset, respectively. The second column shows the result of the original Planet model replicated. The other columns show the performance of the model using the classification approach with patches. It is possible to see that using 5 patches with and without the full image improves the average accuracy by 5 percentage points in Paris dataset and 11 percentage points in Oxford Dataset of the model compared to the initial approach in Paris and Oxford datasets; Using patches strategy allows the classifier to look at small regions that contain information relevant to classification rather than trying to aggregate all the information contained in the image to predict. The regions without relevance will not have much influence on the final result being classified with low confidence value.

In all models for the Paris dataset evaluation the best result

TABLE IV
RESULTS OF CLASSIFICATION FOR EACH CLASS IN PARIS DATASET VARYING NUMBER OF PATCHES AND USING PWR VOTE SCHEME

Paris dataset				
	PlaNet	Patch PlaNet (4 Patches)	Patch PlaNet (5 Patches)	Patch PlaNet (6 Patches)
Eiffel Tower	97.05%	93.13%	96.07%	**98.03**%
La Dfense	89.58%	95.31%	95.83%	**95.85**%
Louvre	89.76%	91.73%	95.28%	**95.67**%
Notre-Dame	94.11%	92.81%	97.00%	**97.60**%
Panthéon	47.45%	42.35%	52.94%	**54.50**%
Sacré-Cœur	86.51%	86.04%	93.02%	**96.28**%
Arc de Triomphe	89.49%	87.89%	**95.85**%	77.92%

TABLE V
RESULTS OF CLASSIFICATION FOR EACH CLASS IN OXFORD DATASET VARYING NUMBER OF PATCHES AND USING PWR VOTE SCHEME

Oxford dataset				
	PlaNet	Patch PlaNet (4 Patches)	Patch PlaNet (5 Patches)	Patch PlaNet (6 Patches)
All Souls	68.83%	62.33%	75.32%	**77.92**%
Ashmolean	86.67%	**96.67**%	**96.67**%	**96.67**%
Balliol	48.00%	**68.00**%	60.00%	**68.00**%
Bodleian	75.30%	80.24%	**82.71**%	80.25%
Christ Church	58.54%	52.57%	65.38%	**67.95**%
Cornmarket	75.00%	90.00%	90.00%	**95.00**%
Hertford	78.78%	78.78%	**81.81**%	**81.81**%
Keble	79.41%	79.41%	**94.11**%	91.18%
Magdalen	61.15%	51.80%	60.43%	**69.78**%
Oriel	38.23%	41.11%	**41.20**%	41.18%
Worcester	44.44%	52.78%	61.11%	**66.67**%

was obtained for the Eiffel Tower, where there was great variety of data in the training set. The worst performance was relative to the Panthéon. The test set for Panthéon contains several internal images and specific regions of the location that training not fully encompasses. The training set contains some characteristics of the interior region, but it has predominantly outdoor images of the class, showing that the model can generalize the classification well for some results, but it is quite dependent on having large amounts of data from places with a high variety of examples.

In the Oxford dataset the results are also better when the number of patches increased. The performance of the model for Oxford is worse when compared to Paris due to the large number of indoor examples and where the landmark has a high level of occlusion, this makes it difficult to evaluate the dataset for some classes as in the case of Balliol, Oriel and Worcester.

In the implementation code, the model can be loaded in 1 second. After loaded, each classification spends $300ms$ per image on average. The file with the graph and parameters trained has a storage occupation of $85MB$. Using the patch recognition strategy, the inference time increases proportionally to the number of patches used. It was seen that using this strategy we can increase the accuracy of the model, but we also decrease the runtime performance of the inference. To achieve an accuracy improvement of 5-11 percentage points it may be necessary to add a running cost up to 6 times, increasing the runtime from $300ms$ to $1.8s$ on CPU.

C. Qualitative Results

We can see in Figure 5 some interesting success cases, where there are cases of partial occlusion, color and brightness changes, soft lack of focus and scale variation.

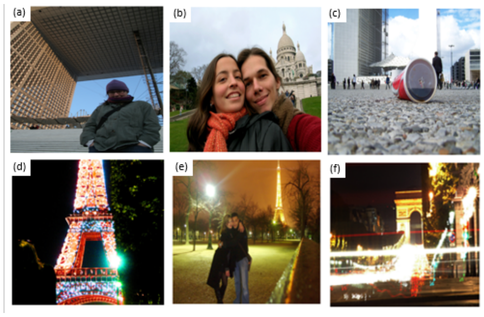

Fig. 5. Success cases. Figure (a) shows the robustness of the model for generalizing knowledge to similar parts of similar regions in the landmarks. In (b) we have the correct recognition in a case of landmark partial occlusion. In (c) the landmark is correctly recognized in the background. In (d) the model recognizes the landmark with high illumination change. (e) shows a case of robustness to moderate scale variation and (f) shows a sample with strong occlusion, illumination and brightness changes that was correctly classified as Arc de Triomphe.

The examples in which PlaNet can classify correctly are also classified correctly by the PlaNet Patch. But using patches we can get more cases with correct predictions.

The patch approach can handle better cases of partial occlusion. When dividing the image, the region that overlap the landmark will be classified with low confidence value and consequently will not influence much in the final decision, whereas the area in which the landmark appears will have greater weight in the classification with high confidence value.

Another advantage in using patches is when we deal with landmarks that have very similar characteristics. The model can better distinguish these cases by looking at regions where they contain small features that differ one landmark to another.

Patches also help in cases where the landmark is on a small scale in relation to the entire image. In cases where is possible to divide the landmark so that it is present partially in a patch, the classifier can recognize the learned features without much information of distraction. The problem in this case is when landmark information has a small scale and the landmark features is split between several patches, which worsens the model's accuracy. The overlap (5 patches and 6 patches) approach can circumvent this problem by preserving key features sharing the information between adjacent patches and minimizing the effect of division.

Figures 6 and 7 show some cases of model failures. The cases shown are: lack of focus on the landmark, very small regions, new information in the context of the image, very low lighting conditions and correlation between classes features.

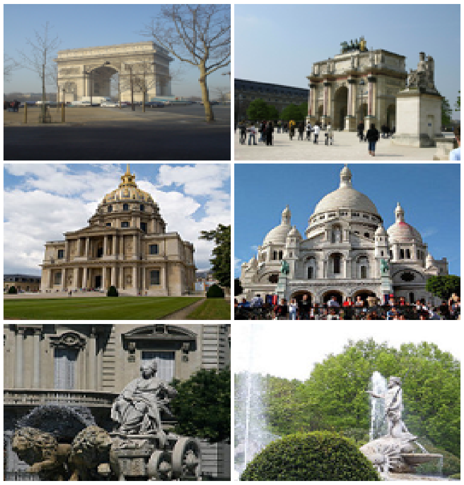

Fig. 7. Other failure cases occur when there is a great correlation between the features of different classes. This failure causes confusion for the model to make a decision. The images on the left correspond to ground truth landmarks and the images on the right correspond to the landmark class predicted by the model.

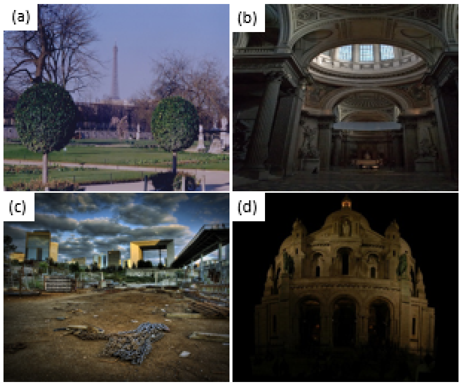

Fig. 6. Failures cases. In (a) the model may misclassify cases where the landmark is too distant (scale problem) and out of focus in the image. In (b) the model cannot recognize the landmark when the area of the landmark in the input image is very different from the samples in the training set (the indoor test sample does not have the same features of outdoor images used in training). (c) shows the dependence of the model in relation to the context of the image. The model can usually handle this problem but fails in cases where the context information is very different from the known ones. In (d) it is seen a failure with very low illumination conditions.

The patches approach does not help in cases where we have several different information in the context around the landmark and the landmark dimensions are rather small in relation to the image. When the division is made, we will have a lot of irrelevant information not included in the training and little information learned about the landmark that will be divided between the defined regions. In these cases the use of original PlaNet and 5 patches + full image performs better on using only the patches.

VI. Conclusion

We propose a new classification strategy for the PlaNet landmark recognition algorithm, in which we could train the original model, modify it under the same conditions and compare its results. We obtain improvements in the accuracy performance increasing on average 5 to 11 percentage points in the classification result .

The strategy of using patches reduces ambiguity by having the classifier look at small overlapping regions to retrieve the information, allowing regions with non-relevant information or noises to be avoided, being classified with low confidence.The best configuration could be obtained by using the patch information and the full image. When the input image contains a lot of noise (objects of distraction, change of lighting, people) the use of the patches allows an improvement in the performance making these noisy areas have low confidence and handling the predictions separately. When the classifier has high confidence that the landmark is present, it can ignore the patch information and return the prediction directly. However, to achieve this improvement, the model loses in computational performance, which can increase its execution time by up to 6 times.

We were also able to create a dataset and validate the procedure to perform data augmentation and model training when using a small dataset. We achieved good acuracy when compared to [5], [12] classifying outdoor and landmark-focused images; Cases that represent the training set used.

For the cases of indoor images, low illumination or small scale our model tends to fail. We have seen that there is a great dependence of the DCNNs models with respect to the data used in the learning phase, being necessary a large set of relevant images in order to generate a good set of features. To improve our accuracy in these cases, we can increase the training set by inserting these types of data and others images transformations. Another possible approach is to test model performance for new DCNN architectures.

Acknowledgment

The authors thank the CNPq (process #132297/2017-5) for the financial support.

References

[1] J. Hays and A. A. Efros, "im2gps: estimating geographic information from a single image," in *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2008.

[2] Y. Kalantidis, C. Mellina, and S. Osindero, "Cross-dimensional weighting for aggregated deep convolutional features," in *European Conference on Computer Vision*. Springer, 2016, pp. 685–701.

[3] E. Mohedano, K. McGuinness, N. E. O'Connor, A. Salvador, F. Marques, and X. Giro-i Nieto, "Bags of local convolutional features for scalable instance search," in *Proceedings of the 2016 ACM on International Conference on Multimedia Retrieval*. ACM, 2016, pp. 327–331.

[4] A. Babenko and V. Lempitsky, "Aggregating local deep features for image retrieval," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1269–1277.

[5] N. Vo, N. Jacobs, and J. Hays, "Revisiting im2gps in the deep learning era," in *2017 IEEE International Conference on Computer Vision (ICCV)*. IEEE, 2017, pp. 2640–2649.

[6] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, "Netvlad: Cnn architecture for weakly supervised place recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 5297–5307.

[7] T. Weyand, I. Kostrikov, and J. Philbin, "Planet-photo geolocation with convolutional neural networks," in *European Conference on Computer Vision*. Springer, 2016, pp. 37–55.

[8] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.

[9] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Lost in quantization: Improving particular object retrieval in large scale image databases," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

[10] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Object retrieval with large vocabularies and fast spatial matching," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2007.

[11] Y. Avrithis, Y. Kalantidis, G. Tolias, and E. Spyrou, "Retrieving landmark and non-landmark images from community photo collections," in *Proceedings of the 18th ACM international conference on Multimedia*. ACM, 2010, pp. 153–162.

[12] G. Tolias, R. Sicre, and H. Jégou, "Particular object retrieval with integral max-pooling of cnn activations," *arXiv preprint arXiv:1511.05879*, 2015.

[13] T.-Y. Lin, S. Belongie, and J. Hays, "Cross-view image geolocalization," in *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*. IEEE, 2013, pp. 891–898.

[14] T.-Y. Lin, Y. Cui, S. Belongie, and J. Hays, "Learning deep representations for ground-to-aerial geolocalization," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 5007–5015.

[15] S. Workman, R. Souvenir, and N. Jacobs, "Wide-area image geolocalization with aerial reference imagery," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 3961–3969.

[16] A. S. Razavian, J. Sullivan, S. Carlsson, and A. Maki, "Visual instance retrieval with deep convolutional networks," *ITE Transactions on Media Technology and Applications*, vol. 4, no. 3, pp. 251–258, 2016.

[17] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. D. and2 Sanjay Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, software available from tensorflow.org. [Online]. Available: https://www.tensorflow.org/

[18] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2818–2826.

[19] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2015.

[20] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE, 2009, pp. 248–255.

[21] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *Journal of Machine Learning Research*, vol. 12, no. Jul, pp. 2121–2159, 2011.

[22] Ludicorp, "Flickr: Togheter," 2004, image and Video hosting service. [Online]. Available: https://www.flickr.com

[23] F. Radenović, G. Tolias, and O. Chum, "Cnn image retrieval learns from bow: Unsupervised fine-tuning with hard examples," in *European Conference on Computer Vision*. Springer, 2016, pp. 3–20.

[24] A. Sharif Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, "Cnn features off-the-shelf: an astounding baseline for recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2014, pp. 806–813.

[25] J. Yue-Hei Ng, F. Yang, and L. S. Davis, "Exploiting local features from deep networks for image retrieval," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2015, pp. 53–61.