Fully Convolutional Network and Region Proposal for Instance Identification with Egocentric Vision

Maxime Portaz, Matthias Kohl, Georges Quénot, Jean-Pierre Chevallet

Fully Convolutional Network and Region Proposal for Instance Identification with egocentric vision

Maxime Portaz
Univ. Grenoble Alpes, CNRS, LIG,
F-38000 Grenoble France
papers@maximeportaz.fr

Matthias Kohl
Univ. Grenoble Alpes, CNRS, LIG,
F-38000 Grenoble France
mattskohl@gmail.com

Georges Quénot
Univ. Grenoble Alpes, CNRS, LIG,
F-38000 Grenoble France
georges.quenot@imag.fr

Jean-Pierre Chevallet
Univ. Grenoble Alpes, CNRS, LIG,
F-38000 Grenoble France
jean-pierre.chevallet@imag.fr

Abstract

This paper presents a novel approach for egocentric image retrieval and object detection. This approach uses fully convolutional networks (FCN) to obtain region proposals without the need for an additional component in the network and training. It is particularly suited for small datasets with low object variability. The proposed network can be trained end-to-end and produces an effective global descriptor as an image representation. Additionally, it can be built upon any type of CNN pre-trained for classification. Through multiple experiments on two egocentric image datasets taken from museum visits, we show that the descriptor obtained using our proposed network outperforms those from previous state-of-the-art approaches. It is also just as memory-efficient, making it adapted to mobile devices such as an augmented museum audio-guide.

1. Introduction

We propose to enhance a museum audio-tour guide with a camera, in order to help user orientation, enable automatic guidance and facilitate museum artifact explanations: when the visitor is close enough to an object, an explanation is automatically launched. The embarked camera is not used for augmented reality but only for the system to localize the user without the need of any extra hardware to be installed in the museum (a very strict constraint in some museums). Hence, only egocentric image analysis and object instance recognition is possible to localize the user on the museum map. The camera is installed on a small device held in the user's chest. Thus, this system needs to recognize object instances (not classes), such as paintings, sculptures, or any exhibited historical heritages. Obviously, the entire museum must be photographed (or video recorded), and each object image then has to be manually localized in the digital museum map.

Instance search is a visual task that aims, given an image, to identify the particular objects shown. It is different from Image Classification, where we focus on identifying object category, with robustness to intra-class variability. It is also different because of the nature of the data. An instance represents a given object, generally described only by a few shots. The instance search task is similar to Image Retrieval, where the aim is to retrieve all images that contain the same object instance from a query image. Instance search uses the result of the image retrieval system to identify object instances.

We propose a system that learns image representations and allows Instance Identification through image retrieval. We use Deep Learning Neural Networks in order to learn this representation. The Neural Network model proposed is learned with a siamese network with three streams and a triplet loss [29]. The aim is to train the weights inside the network to produce an image representation that allows image comparison based on their contents. Because of the number of images available in Instance Search dataset, we need an external source of data to train a convolutional network, such as ImageNet [27].

The network we use is a Fully Convolutional Network (FCN) [17] that allows any input size, to avoid image deformation or scaling. Additionally, the FCN can be used to produce region proposals without any additional component, in the network or in the dataset. For the training phase, we use the triplet loss between the three streams of the siamese architecture, and a cross entropy loss for classification of the region with the highest activation. The aim is to create a representation of the image that captures the position of the object and the difference between images, whether similar or not.

At test time, the trained FCN is passed over the whole image, but only the location with the top k maximal activations will form the image description. This representation is compared with the reference images in the dataset using a dot product, to obtain the closest reference image representing the instance.

In order to evaluate our approach, we use two egocentric datasets from museum visits [25]. These datasets represent instances with only a few examples and with very little variability. We show that our approach achieves better results than the previous state of the art by Gordo et al. [12].

We first present in section **??** the related work on instance identification. Then, in section 3 we describe why the use of pre-trained CNNs and fine-tuning is important for our problem. The section 4 presents the region of interest detection and object localization. The section 5 describes the proposed network and how it was trained on the datasets we used. Finally, experimental results and evaluation are shown in section 6.

2. Related Work

Before the ground-breaking results of deep learning methods for object detection and image retrieval, shallow patch descriptors have been used in several domains. The SIFT [18] descriptor was the most used one, among the large variety of traditional patch descriptors. It has been successfully employed for tasks like image search with content based retrieval [14] or classification [21]. For image retrieval, methods inspired by text retrieval methods, such as bag of words [6], used bag-of-features (BoF) image representation [32]. Aggregated descriptors like VLAD [15] or Fisher Vectors [21] are an evolution of BoF, with smaller vocabulary, more adapted to large datasets. The advantages of traditional patch descriptor approaches is the possibility of spatial verification and geometry verification to improve object retrieval [20, 22], and the possible combination with methods like query expansion [7, 2].

Starting with the results of AlexNet for image classification in the 2012 ImageNet challenge [16, 27], image classification tasks have been dominated by CNNs. A CNN trained on a large enough labeled dataset like ImageNet can be used as a feature extractor with its intermediate layers, to construct an image representation for image retrieval [4, 30]. To overcome the lack of geometry invariance of this approach [10], cross-matching [30], sum-polling [3] or fine-tuning [4] with an external dataset can be used.

In order to compare images for image retrieval, image patch comparisons have shown better results than SIFT [9, 31, 36]. Image patches can be constructed with deep patch descriptors [9] as patch label, each patch is a label, by learning patch differences with a siamese network [31, 36], or with a Convolutional Kernel Network [19].

Another important aspect of image retrieval is to learn to rank [11, 1]. While Arandjelovic et al. [1] have shown the importance of learning to rank, Gordo et al. [11] used a siamese network along with a triplet loss, previously used for face recognition [29], to construct an effective image representation by learning with a similarity metric. The work presented in this paper follows this idea.

3. CNN Fine-tuning

The main difference between image classification and image retrieval, is the amount of data and their variability. In classification, we rely on a large amount of data, with high variability of examples for each category. We can then train a Deep Convolutional Network with millions of parameters. In image retrieval or identification, as we want to identify a particular instance, the variability of examples is less important, and not sufficient to train a network like ResNet. In order to use a CNN, we only fine-tune a CNN pre-trained on a bigger collection for image classification. Fine-tuning focuses on the higher layers of a CNN and can increase generalization even in the fine-tuned model[35]. We focus our work on two different well studied models of CNN, AlexNet [16] and ResNet [13].

3.1. Transfer learning

The modularity of a CNN means that we can easily transfer the weights from a pre-trained model, and only re-train the highest abstraction layers. Specifically, we re-train all linear layers in the model, representing the highest-level layers.

We also re-train the highest level convolutional layers, since our datasets contain many visually different images as compared to the ImageNet dataset used for pre-training the models. For the AlexNet architecture, we choose to re-train all layers above and including the last convolutional layer. For a ResNet architecture, we re-train all layers above and including the third to last block of convolutional layers. This contains the nine highest convolutional layers in total.

3.2. Data Augmentation

Image retrieval methods focus on problems with few examples and little variability in instance images. This leads to too few data to train a typical CNN model designed for classification, even with fine-tuning. One way to overcome this is to augment the data, by randomly applying affine transformations, color perturbations and other random transformations.

The lack of geometry invariance and scaling invariance of the model can be reduced by randomly rotating and flipping the images and using different scaling, thus we perform this type of data augmentation throughout our experiments.

For data augmentation in order to fine-tune a CNN, we use the following values in our experiments:

1. Rotation: any angle is chosen with the same probability.

2. Scaling: the scaling factor is chosen independently for each dimension in the range $[0.75, 1.25]$.

3. Flipping: with probability 0.5, images are horizontally flipped.

4. Region of Interest and object Localization

Previous approaches in image retrieval [12, 28, 33] usually deal with regions of interest in one way or another. The idea is that in most cases, only certain parts of each image can be useful for comparison with other images. In addition to this, cropping images at their regions of interest can help with differences in scale of the images to compare: if a painting is visible only in a small part of an image, cropping the image at that part and then re-scaling the part should set the painting at a normalized scale.

However, in instance search with museum datasets, it is not obvious where the regions of interest should be: most images represent an entire painting or parts of it and only some may contain the painting as part of the image with a wall in the background. This means for most images, the ground-truth region of interest is simply the entire image, and some may have a ground-truth region of interest which is almost the entire image, excluding only a small part of the background.

On the other hand, a network fine-tuned on classification on such a dataset should be able to easily identify the region containing the painting, since the background wall is contained in almost all classes, which means it is a particularly bad indicator of the class. Thus, if the network is applied in a strided manner across an image, it should produce low maximal activations in parts containing big sections of background wall.

Figure 1 shows images, along with the heat map representing the maximal activation of a fine-tuned ResNet-152 at each coordinate, when the network is applied in a strided manner across the input image. From this image, we can see that the highest maximal activations of the network usually occur at the location of the object. This is true even if the object is not correctly classified by some of the highest activations as can be seen in the second image.

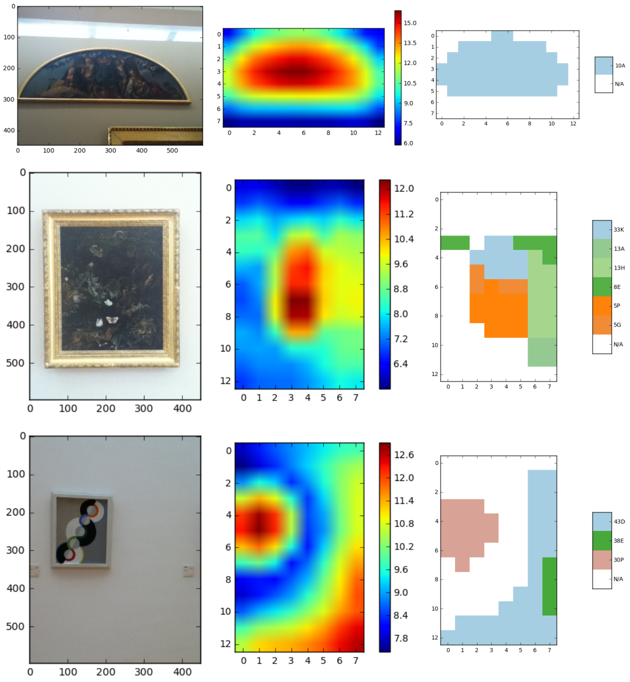


Figure 1: Sample images (scaled to a smaller side of 448 pixels) along with the heat-map of maximal activation values at each location when a fine-tuned ResNet-152 is applied to the image in a strided manner, as well as the labels of all maximal activations that are greater than the mean maximal activation

In the third image, it seems like many high maximal activations occur specifically in the background area. However, the corresponding label-map shows that these areas correspond to the labels 38E and 43D. Both of these labels are pieces of art which consist mostly of the background wall. In this sense, it is not entirely wrong to consider 'wall-only' patches of the image as instances of these pieces of art. This simply means that the image consists of two separate regions of interest: one region with the painting (label 30P) and one region with the wall (labels 38E/43D).

From these observations, we can confirm the assumption that the maximal activations of a fine-tuned network are a good indicator of the location of an object, or a combination of different objects. Using this assumption, there is no need for a procedure to annotate regions of interest, as employed by most state-of-the-art image retrieval approaches [11, 33, 26].

On the other hand, using datasets developed for image retrieval, such as Paris6k or Oxford5k [24, 23], this assumption cannot be applied, since the dataset is not clean enough for a network fine-tuned on classification to be a good indicator of location of the query objects.

5. Fine-tuning on classification using FCN

As shown before, a fine-tuned CNN is already a good indicator of the location of an object in our datasets. Additionally, it seems like scale is a particularly important factor.

5.1. Fully Convolutional Network

Thus, the idea is to start by fine-tuning a network with images at different scales. This can be achieved by using a fully convolutional network (FCN) [17].

In an FCN, the final fully connected layers of a network are replaced by convolutional layers having a kernel which fits the entire domain of the output of the previous layer. This type of convolution is equivalent to a fully connected layer, but allows inputs (and outputs) of any size. The effect is that the network can be applied in one pass to an arbitrarily sized image. The output then represents the activations of the network as if it was applied in a strided manner across the image. The stride of a full network depends on the architecture and is 32 pixels for the architectures used here: AlexNet and ResNet.

Once an FCN is applied to the image, the loss is calculated by averaging the cross-entropy (CE) loss (eq. 1).

$$\mathcal{L} = \frac{1}{N}\sum_{i=1}^{N} CE_i \tag{1}$$

The final loss is then obtained by passing images at different scales through the FCN and averaging across all cross-entropy losses of all outputs and scales (eq. 2).

$$\mathcal{L} = \frac{1}{S}\sum_{s=1}^{S}\frac{1}{H_s * W_s}\sum_{h=1}^{H_s}\sum_{w=1}^{W_s} CE^{h,w} \tag{2}$$

In equation 2, S represents the number of scales at which the image is passed through the network. H_s and W_s represent the height and width of the feature map respectively, given the scale s of the input image. $CE^{h,w}$ represents the cross-entropy of the network at spatial location (h, w) in the full feature map.

We choose to give each scale of the image the same weight in the loss. This is because the images are passed to the network at their true aspect ratio, which means the loss for different images may have different values for the heights and widths of the feature maps H_s and W_s.

In order to normalize the sizes of the features present in the images, all images are scaled to have the same number of pixels in the smaller side. Note that for large aspect ratios and large scales of the smaller side, the memory consumption of training can be high for single images having a very large aspect ratio. To limit this spike in memory consumption, the aspect ratios are limited by introducing uniform random noise on the smaller side of images with high aspect ratios.

In our experiments, we use a maximal aspect ratio of 2.0 and images at two scales of 448 and 224 pixels for the smaller side. We found that the AlexNet architecture did not have good convergence behavior, thus we used scales of 384 and 224 instead.

5.2. Triplet Selection

The network is trained with image triplets, with a siamese configuration. The loss used for this training is the triplet loss [29]. Equation 3 shows the triplet loss for N images with a being the anchor image (query), n the negative example and p the positive one. The typical triplet loss is defined using squared distances. For normalized vectors, we can express it using the dot product. This leads to simpler gradient computation. In equation 3, $x_i^a x_i^n$ corresponds to the similarity between the anchor and the negative example and $x_i^a x_i^p$ to the similarity between the anchor and the positive example. The scalar m represents the margin between a positive and a negative pair of images.

$$\mathcal{L} = \sum_{i=1}^{N} \max(0, x_i^a x_i^n - x_i^a x_i^p + m) = \sum_{i=1}^{N} \mathcal{TL}_i \tag{3}$$

As noted by previous authors, when using the triplet loss, it is crucial to choose the best triplets during training in order to obtain convergence. In particular, many triplets are irrelevant and do not produce any loss since they are *too easy* for the network.

Hence, the first idea is to choose the hardest triplets, as proposed by Schroff et al [29]. However, as they show in their paper, this can lead to a collapsing model early on in training. Thus, they choose *semi-hard* triplets instead. Semi-hard triplets are obtained as follows: use all possible positive couples of images (couples of images from the same instance). For each positive couple, choose the hardest negative that is easier than the positive couple. Hard and easy are defined by the dot product between the descriptors of the images: a high value of the dot product for images of the same instance represents an easy positive couple, a high value of the dot product for images of different instances represents a hard negative couple. The value of all dot products are determined before each pass over the whole training data during training, for all couples of images.

A different triplet selection mechanism was proposed by Gordo et al. [12]. First, calculate the values of dot products for all couples of images before each pass over the training data. Second, for each image, choose the n easiest positive images and the m hardest negatives. Then, calculate the loss for all possible combinations and use the o triplets with the highest loss. This method probably eliminates some noise when choosing the easiest positive couples, for images that are labeled as being the same instance but are not visually similar. However, in experiments, we found that this method does not perform well for datasets with few images per instance, since we either have to choose n as very low or we end up choosing all positive couples after all for most instances, just like in the *semi-hard* selection.

Hence, in our experiments, we choose the semi-hard triplet selection for the first two passes over the dataset, after which we only choose the hardest negatives for all positive couples.

5.3. Descriptor Extraction Network

Figure 2 illustrates the proposed architecture. To obtain a descriptor, we first apply the convolutional layers of a previous architecture, such as AlexNet or ResNet. We then obtain all classification outputs at all locations using the FCN. We only consider the maximal activation at all locations. The locations with the top k maximal activations will form the descriptor.

For each of these locations, the convolutional features are reduced by a $\| \cdot \|_2$-normalization, then a shifting and fully connected layer. Finally, all descriptors from the k locations are sum-aggregated and $\| \cdot \|_2$-normalized again.

When training, the network is applied to a triplet of images. These triplets are chosen as described in Section 5.2.

Additionally, we regularize the triplet loss by a cross-entropy loss to make sure that the k locations with highest maximal activations are correctly classified. This loss is averaged over the k locations.

Equation 4 shows the full loss as used in our experiments to train the proposed model, for N images. In this equation, (h_l, w_l) represent the spatial coordinates of the l-th region of highest maximal activation in the feature map produced by the FCN.

$$\mathcal{L} = \sum_{i=1}^{N} \left(\mathcal{TL}_i + \alpha \frac{1}{N} \frac{1}{k} \sum_{l=1}^{k} CE_i^{h_l, w_l} \right) \tag{4}$$

In our experiments, we choose the number of regions with highest maximal activation to be $k = 6$ and the regularization hyper-parameter $\alpha = 1.0$. The margin of the triplet loss is $m = 0.1$.

This approach allows the network to decide which region of interest is best suited for classification and ultimately which regions are best suited for comparison with other images. Another advantage is that this approach does not require any annotation of the images with regions of interest, which can be a long, manual or automatic process, as evident from the cleaning process used by Gordo et al [12].

Finally, an important property of the descriptor is that it heavily relies on the classification capabilities of the network. This means the descriptor is mostly meaningless for a different dataset and needs to be learned for each dataset. This can be an advantage, since the descriptor can be better suited to a particular dataset and the learning process does not take long. On the other hand, it means that the descriptor cannot be applied in a typical image retrieval task.

5.4. Instance Feature Augmentation

Query Expansion [8] in Image Retrieval, using deep learning, like shown by Gordo et al [12], is possible and relies on a combination of the image descriptor and the descriptors of the top k retrieve results. This new descriptor is used to perform a second query, which gives the final result.

Furthermore, we do not expect query expansion to provide any major improvements in our research problem, since we expect to have very few images returned. This means the only plausible value of k would be $k = 1$. However, if the best matching descriptor of the first query already matched, the second query cannot improve the result and if it does not match, it is unlikely that the second query would match.

An approach called Database-side feature augmentation [34, 2], proposes to combine descriptors of the reference images in order to form better database-side descriptors. Every reference descriptor is simply replaced by a combination of itself and the k nearest neighbors. This combination is computed as a weighted sum, weighted by the rank of the neighbors with respect to k (the closest neighbor has the highest weight and the k-th neighbor the lowest).

In our work, we use a technique called Instance Feature Augmentation. We use the fact that we know the corresponding label for each image in our dataset. For each label, we compute the representation of an instance by averaging the features of every images corresponding to this label. This representation is added to the dataset as a new instance. We show that this approach does not improve mean precision@1, but gives a better Mean Average Precision. This suggests that the internal representation of the instance is improved.

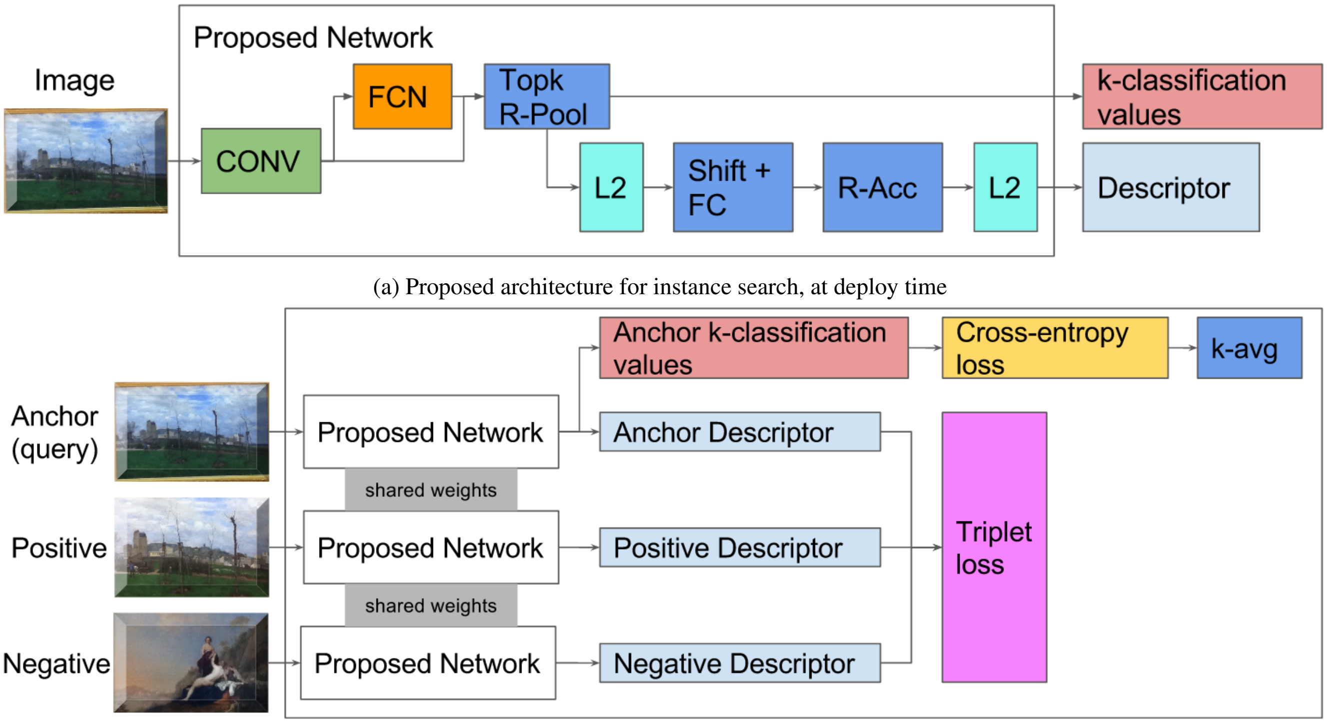
6. Evaluation

6.1. Datasets

The proposed approaches as well as several base-lines are evaluated on two datasets: the CLICIDE and GaRoFou datasets. These datasets are described in detail by Portaz et al. [25], and they represent artwork photos, taken by classical or head-mounted cameras. Both datasets are typical of instance search datasets in museums or touristic sites with egocentric view: the objects represented by their images are paintings for one and glass cabinets containing sculptures and artifacts for the other dataset. Both datasets contain a small number of images per instance and a small number of images in total.

6.2. Results

Table 1 gives an overview of the results obtained. First, the baselines established by SIFT descriptor, CNN network features extraction are shown. Additionally, we show the relevant results obtained by fine-tuning a classification network, abbreviated by FT in the table. We then show the results obtained by a simplified Siamese architecture, ab-



(a) Proposed architecture for instance search, at deploy time

(b) Proposed architecture for instance search, at training time

Figure 2: Proposed architecture for instance search, based on an FCN [17] for region proposals.

	Mean Precision@1		*Mean Average Precision*	
	CLICIDE	*GaRoFou*	*CLICIDE*	*GaRoFou*
SIFT	70.08	78.82	N/A	N/A
Gordo et al. [11] (ResNet-50)	90.30	95.65	65.49	88.43
Gordo et al. [11] (ResNet-50, multi-res)	92.73	95.65	N/A	89.32
AlexNet IN	72.73	85.87	32.71	66.11
AlexNet FT	78.18	90.76	38.51	72.92
AlexNet SS	75.76	90.20	36.20	77.73
Proposed AlexNet	81.21	83.15	45.53	71.71
Proposed AlexNet (IFA)	80.61	82.61	71.02	81.66
ResNet-152 IN	72.12	85.33	40.99	70.15
ResNet-152 FT	79.39	94.57	75.11	93.44
ResNet-152 SS	85.45	95.11	83.00	91.90
Proposed ResNet-152	**94.55**	**96.20**	82.94	91.83
Proposed ResNet-152 (IFA)	93.94	95.11	**94.23**	**93.86**

Table 1: Evaluation results for the CLICIDE and GaRoFou datasets. The results are expressed in percentage points of mean precision@1 and mean average precision (only indicative)

breviated SS. Finally, we show the results obtained by the proposed network. In addition to the mean precision@1, we show the mean average precision obtained by the different approaches.

From the baselines presented, we can make two observations. First, even a simple global descriptor obtained from the convolutional features of a CNN pre-trained on ImageNet performs better than matching local SIFT descriptors on our datasets. Second, the ResNet-50 proposed by Gordo et al. [11] out-performs the descriptors from pre-trained networks by far, even though it has never seen the images from our datasets during training, either.

Table 1 confirms these observations when taking into account the mean average precision of the ResNet-50 and the convolutional features of networks pre-trained on ImageNet. The difference is more than 10 points gained in mean average precision even when comparing against the ResNet architecture. This means that a ResNet fully optimized for image matching captures the visual information much better than just the convolutional features of a pre-trained network. This is expected, since that was one of the goals of the approach proposed by Gordo et al. [11].

Another observation we can make from Table 1 is that fine-tuning a network on the reference dataset consistently out-performs a pre-trained network. This shows that transfer learning is very powerful for small datasets with many classes. Indeed, networks with many parameters such as AlexNet and ResNet could not have been trained on such small datasets with uninitialized weights.

However, when comparing the classification fine-tuning method with the simplified Siamese architecture (fine-tuning with a triplet loss), it is not as clear which one performs better. From the results, we can see that the classification fine-tuning has a better performance for AlexNet while the triplet loss fine-tuning has a better performance for ResNet-152. This is most likely due to two factors: the hyper-parameters when training the Siamese AlexNet were not perfectly suited, hence the convergence behavior is not as good as with the Siamese ResNet. Furthermore, the AlexNet fine-tuned for classification has a much larger descriptor of dimension 9216 versus the descriptor of dimension 2048 of the simplified Siamese architecture. This may explain that the simplified Siamese architecture performs worse in this case.

Finally, when comparing the proposed architecture with the previous ones, it is clear that the proposed architecture out-performs all of them. It achieves higher precision@1 as well as higher mean average precision, especially when combined with the instance feature augmentation. The comparison with the ResNet-50 from Gordo et al [11] is difficult though. This is because on the one hand, our proposed network is trained on the reference dataset used when comparing images, giving it an unfair advantage. On the other hand, the ResNet-50 is trained on the much larger Landmarks dataset [5], giving it the advantage of data volume. The training methodology developed by Gordo et al. is not applicable to a small, clean dataset, such as the ones used in our evaluation.

6.3. Limitations and Future work

We tested our approach on museum datasets, only on still images, with and without egocentric point-of-view. For these approaches to work on videos, we need to adapt them. Early tests suggest that a different training is mandatory, to take into account the specificity of video (motion blur, obstruction, . . .).

The GaRoFou dataset is made to test the system with cluttered scenes, as several objects can be seen at the same time. However, we could not test the system's robustness to crowded scenes, or to obstruction. Finally, more realistic and more challenging corpora are required for further training as well as testing.

7. Conclusion

This paper presents a novel approach for instance and image retrieval with low variability and small datasets. The proposed approach consists of two key steps. First, we leverage the concept of fully convolutional networks in order to perform classification training at different scales, without a heavy computational overhead. Second, we show that the fully convolutional network can be used to obtain region proposals without the need for an additional component in the network and training. This is particularly important, since region proposals are costly to define manually in our research problem. The region proposals used by the state of the art do not seem applicable to that kind of problem of instance search.

Finally, the proposed network keeps all the benefits of state-of-the-art approaches: it can be trained end-to-end and it produces an effective global descriptor, which can be compared using the dot product. Additionally, it is modular in the sense that it can be built upon any type of CNN, pre-trained for classification. Furthermore, the training time for each dataset is reasonable and the proposed network is fast to evaluate, making it particularly useful for an embedded device such as an augmented museum audio-guide.

Through multiple experiments on two datasets, we show that the descriptor obtained using our proposed network outperforms previous state-of-the-art approaches on the instance search task, while being just as memory-efficient and fast for encoding images. The experiments were conducted on two egocentric image datasets taken from museum visits.

8. Acknowledgment

This work was carried out in the context of the GUIMUTEIC project, funded by the Fonds Européen de Développement Régional (FEDER) of région Auvergne Rhône-Alpes.

References

[1] 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*, pages 5297–5307, 2016.

[2] R. Arandjelović and A. Zisserman. Three things everyone should know to improve object retrieval. In *Computer Vision*

and Pattern Recognition (CVPR), 2012 IEEE Conference on, pages 2911–2918. IEEE, 2012.

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

[4] A. Babenko, A. Slesarev, A. Chigorin, and V. Lempitsky. Neural codes for image retrieval. In *European conference on computer vision*, pages 584–599. Springer, 2014.

[5] A. Babenko, A. Slesarev, A. Chigorin, and V. Lempitsky. Neural codes for image retrieval. In *European conference on computer vision*, pages 584–599. Springer, 2014.

[6] L. A. Barroso, J. Dean, and U. Holzle. Web search for a planet: The google cluster architecture. *IEEE micro*, 23(2):22–28, 2003.

[7] O. Chum, J. Philbin, J. Sivic, M. Isard, and A. Zisserman. Total recall: Automatic query expansion with a generative feature model for object retrieval. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8. IEEE, 2007.

[8] O. Chum, J. Philbin, J. Sivic, M. Isard, and A. Zisserman. Total recall: Automatic query expansion with a generative feature model for object retrieval. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8. IEEE, 2007.

[9] P. Fischer, A. Dosovitskiy, and T. Brox. Descriptor matching with convolutional neural networks: a comparison to sift. *arXiv preprint arXiv:1405.5769*, 2014.

[10] Y. Gong, L. Wang, R. Guo, and S. Lazebnik. Multi-scale orderless pooling of deep convolutional activation features. In *European conference on computer vision*, pages 392–407. Springer, 2014.

[11] A. Gordo, J. Almazán, J. Revaud, and D. Larlus. Deep Image Retrieval: Learning Global Representations for Image Search. In *Computer Vision – ECCV 2016*, pages 241–257. Springer, Cham, Oct. 2016.

[12] A. Gordo, J. Almazan, J. Revaud, and D. Larlus. End-to-end learning of deep visual representations for image retrieval. *arXiv preprint arXiv:1610.07940*, 2016.

[13] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[14] H. Jegou, M. Douze, and C. Schmid. Hamming embedding and weak geometric consistency for large scale image search. *Computer Vision–ECCV 2008*, pages 304–317, 2008.

[15] H. Jégou, M. Douze, C. Schmid, and P. Pérez. Aggregating local descriptors into a compact image representation. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 3304–3311. IEEE, 2010.

[16] A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet Classification with Deep Convolutional Neural Networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.

[17] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3431–3440, 2015.

[18] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

[19] M. Paulin, M. Douze, Z. Harchaoui, J. Mairal, F. Perronin, and C. Schmid. Local convolutional features with unsupervised training for image retrieval. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 91–99, 2015.

[20] M. Perd'och, O. Chum, and J. Matas. Efficient representation of local geometry for large scale object retrieval. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 9–16. IEEE, 2009.

[21] F. Perronnin and C. Dance. Fisher kernels on visual vocabularies for image categorization. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. IEEE, 2007.

[22] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. IEEE, 2007.

[23] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. IEEE, 2007.

[24] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Lost in quantization: Improving particular object retrieval in large scale image databases. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.

[25] M. Portaz, J. Poignant, M. Budnik, P. Mulhem, J. Chevallet, and L. Goeuriot. Construction et évaluation d'un corpus pour la recherche d'instances d'images muséales. In *COnférence en Recherche d'Informations et Applications - CORIA 2017, 14th French Information Retrieval Conference. Marseille, France, March 29-31, 2017. Proceedings, Marseille, France, March 29-31, 2017.*, pages 17–34, 2017.

[26] F. Radenović, G. Tolias, and O. Chum. CNN Image Retrieval Learns from BoW: Unsupervised Fine-Tuning with Hard Examples. In *Computer Vision – ECCV 2016*, pages 3–20. Springer, Cham, Oct. 2016.

[27] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision*, 115(3):211–252, Dec. 2015.

[28] A. Salvador, X. Giro-i Nieto, F. Marques, and S. Satoh. Faster R-CNN Features for Instance Search. *arXiv:1604.08893 [cs]*, Apr. 2016. arXiv: 1604.08893.

[29] F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 815–823, 2015.

[30] 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*, pages 806–813, 2014.

[31] E. Simo-Serra, E. Trulls, L. Ferraz, I. Kokkinos, and F. Moreno-Noguer. Fracking deep convolutional image descriptors. *arXiv preprint arXiv:1412.6537*, 2014.

[32] J. Sivic, A. Zisserman, et al. Video google: A text retrieval approach to object matching in videos. In *iccv*, volume 2, pages 1470–1477, 2003.

[33] G. Tolias, R. Sicre, and H. Jégou. Particular object retrieval with integral max-pooling of CNN activations. *arXiv:1511.05879 [cs]*, Nov. 2015. arXiv: 1511.05879.

[34] P. Turcot and D. G. Lowe. Better matching with fewer features: The selection of useful features in large database recognition problems. In *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on*, pages 2109–2116. IEEE, 2009.

[35] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson. How transferable are features in deep neural networks? *arXiv:1411.1792 [cs]*, Nov. 2014. arXiv: 1411.1792.

[36] J. Zbontar and Y. LeCun. Stereo matching by training a convolutional neural network to compare image patches. *Journal of Machine Learning Research*, 17(1-32):2, 2016.