UNIVERSITÉ Grenoble Alpes

Grenoble INP

Master of Science in Informatics at Grenoble
Master Informatique
Specialization Data Science

Deep learning with Siamese networks for instance search or identification

Matthias Kohl

June 20 (TODO), 2017

Research project performed at LIG - MRIM

Under the supervision of:
Georges Quénot, Jean-Pierre Chevallet

Defended before a jury composed of:
Head of the jury (TODO)
Jury member 1
Jury member 2

June 2017

Abstract

Your abstract goes here... TODO

Acknowledgement

I would like to express my sincere gratitude to .. for his invaluable assistance and comments in reviewing this report... Good luck :) TODO

Résumé

Your abstract in French goes here... TODO

Contents

— 1 —
Introduction

1.1 Motivation

The research presented here is motivated by the GUIMUTEIC project, a collaborative project between industry and LIG. The aim is to develop a smart audio-guide for touristic or cultural sites.

In practice, the final product should offer an augmented reality interface to the user, with information about the object or objects the user is looking at.

One part of the development consists in finding ways of identifying objects the user is looking at. There are multiple possibilities, for example based on the geo-localization of the user and other sensors. In our research, we focus on the recognition of objects based only on visual clues.

1.2 Research problem

More specifically, we are interested in the following problem: We are given a collection with reference images for each object, or instance, to be recognized. The task is to develop a system that, given an image of one of the instances, can decide which instance the image represents. We assume that, on average, less than ten images are available for each instance. We will refer to this problem as instance retrieval in the following.

There are a few problems similar to instance retrieval. For one, there is the image classification problem: we are given a collection of images where each image is assigned a class, like dog or fridge. The task is to develop a system that, given an image, decides which class the image represents. This problem is of course similar if we simply consider each instance to be a separate class. However, in classification, we usually assume to have many images per class: hundreds or even thousands. This means our problem is closer to few-shots classification, where only few images are available for each class.

Another similar problem is image retrieval. In image retrieval, we are given a collection of reference images and a query image. We aim to rank the reference images by similarity to the query image. Usually, in image retrieval, reference images are not labeled or loosely labeled and many images are irrelevant to the query image. The challenge is to rank the most similar images on top. Instance retrieval is related to image retrieval in that we aim to develop a notion of similarity between images. However, in instance retrieval, we do not care about the rank of the returned images, since we only consider the highest ranked image, which should represent the instance to retrieve.

1.3 Challenges

For all of the problems described above, deep learning approaches based on convolutional neural networks (CNNs) have recently obtained the state-of-the-art results. One of the drawbacks of CNNs is that they require large amounts of data to be trained.

In our research problem in particular, we do not have large enough amounts of data available to fully train a CNN. However, we aim to develop a system that can learn a descriptor specific to the reference images, as the system only needs to recognize and differentiate between instances of that particular dataset. So the system should be tuned to the dataset. Since we usually have less than ten images per instance, this is an important challenge to overcome.

Another challenge comes from the nature of the datasets: we were not able to find similar datasets in the literature, with only a few, clean images, for each instance and many instances. Common datasets in image retrieval like Oxford5k [14], Paris6k [15] or Holidays [9] contain many images per instance, but a lot of noise, as they are used to evaluate whether a system can robustly retrieve the correct images out of a set of random images. This means that the approaches used for image retrieval datasets may not work as well for instance retrieval.

1.4 Contributions

We made the following three major contributions:

1. We analyze existing approaches to image retrieval, classification and determine their shortcomings in our problem setting.

2. Based on these shortcomings, we propose a novel approach to improve results in our problem setting.

3. We evaluate the novel approach and compare with other approaches.

— 2 —
State of the art

2.1 SIFT and bag-of-words

Until recently, the state of the art in image retrieval and matching was based on the idea of bag-of-words [14] [13]. The idea behind the approach is to represent an image as a histogram, or collection of frequencies, of visual words. The visual words should be small, representative patches of the images in the dataset. Usually, these visual words are obtained in multiple steps. First, local features are extracted and encoded. The most common feature used is SIFT [12], a 128-dimensional vector representing scale-invariant features. Then, the features are extracted for all images and clustered into clusters of similar features. The representative for each cluster is the center of the cluster, i.e. the mean of all features falling into that cluster. Each image can then be represented as a histogram of the occurrences of these representative features. This histogram forms the image descriptor, which has as many dimensions as there are representative features and clusters.

Finally, for image classification, a classifier can be learned from the descriptors of all images in the dataset. On the other hand, for image retrieval or matching, the descriptors can be directly matched against each other, based on some similarity measure. In both cases, it can be useful to project the descriptors into a Hilbert space using a different similarity measure than euclidean distance in the original descriptor space. For classification, this is done by employing an SVM classifier with a non-linear kernel [21]. Among the most popular kernels are the radial basis function [19] and the chi-squared function [24].

Until recently, this approach has obtained state of the art results for image retrieval tasks [13].

2.2 Deep learning with CNNs

Starting with the results of AlexNet for image classification in the 2012 ImageNet challenge [10] [18], image classification tasks have been dominated by CNNs, learned using large amounts of data.

A general trend in image related tasks is to move to an end-to-end approach, where the final objective is directly optimized using gradient descent and the gradient is back-propagated to all previous parts of the system. In contrast, the bag-of-words model requires a choice of features (SIFT, ORB (TODO mention ORB before), . . .), a choice of the method for clustering features (k-means), a choice of the classifier (AdaBoost [2], SVM, . . .), as well as a choice of the kernel if an SVM classifier is used.

Using a CNN, features are extracted at a low abstraction level by the first convolutional layers, then higher level features are formed by combining low level features from the previous layers. Finally, high-level features are combined into a classifier by linear layers. The advantage of this approach is that the features are learned at all abstraction levels. Furthermore, the modularity of the approach allows us to easily transfer the lower level features learned from a large dataset to a smaller dataset, where there may not be enough data to efficiently learn lower level features.

The following sections describe the high-level architecture of the most widely used CNNs in classification and image retrieval.

2.2.1 AlexNet

AlexNet [10] contains five convolutional layers and three linear layers. The first convolutional layer has a large kernel and stride to quickly increase the receptive field of the consequent filters. The first two convolutional layers are followed by a pooling layer to further increase the receptive field.

Finally, an important improvement for AlexNet to avoid over-fit is the addition of dropout layers [7] before the two first linear layers.

Apart from that, AlexNet is a simple adaptation of LeNet by LeCun et al [11] for the ImageNet challenge.

2.2.2 VGG

The VGG architecture, introduced by Simonyan et al [22] is the first to use exclusively 3×3 convolutional kernels. This means that the receptive field of the first filters is much smaller, and thus many more layers are needed, as well as more pooling layers. The most popular VGG architectures are VGG-16 and VGG-19, having 16 and 19 layers in total respectively.

Using smaller kernels in convolutions and consequently more layers allows to reduce the number of trainable parameters used by the network, while increasing the number of non-linear layers. This was shown to allow better generalization properties, and thus less over-fit [22].

2.2.3 ResNet, Inception, DenseNet

The idea of increasing the number of layers is further developed by the ResNet, DenseNet and Inception architectures. However, naively increasing the number of layers leads to the problem of vanishing gradient [6].

All of these networks overcome the vanishing gradient problem by introducing skip connections, where the output of a previous layer is added to the output of a layer, skipping some number of layers in between.

The ResNet architecture by He et al [5] always uses blocks of 3 layers along with a skip of those 3 layers. The smaller types of ResNet use blocks of 2 layers.

The Inception architecture by Szegedy et al [23] uses a combination of skip connections of different length and different types.

Finally, DenseNet by Huang et al [8] takes the skip connection idea one step further: the input of each layer is dependent on the output of all n previous layers to some extent.

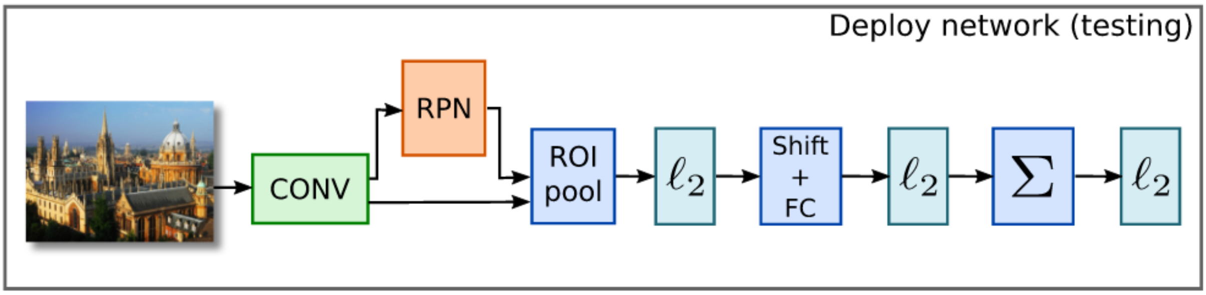

Figure 2.1: Architecture of a CNN network for image retrieval

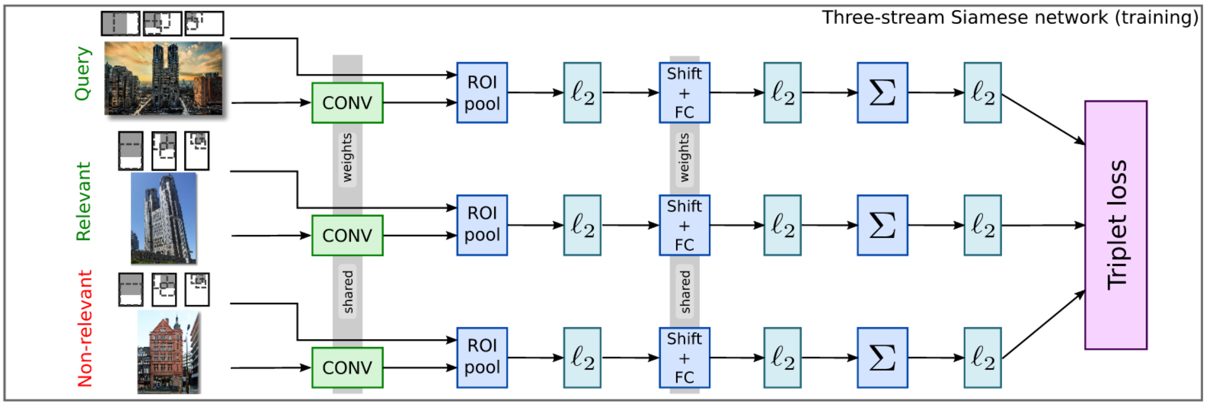

Figure 2.2: Siamese architecture of a CNN for image retrieval used in training

Finally, all of the very deep architectures use batch norm layers to further reduce over-fit: a batch norm layer simply normalizes the features over each batch, and then applies a learnable scaling and shifting.

2.3 Image retrieval using CNNs

For image retrieval, the current state of the art is set by Gordo et al [3]. It is based on an end-to-end approach. The goal is to learn a global descriptor for images that is well suited for comparing images.

Figure 2.1 shows the detailed architecture used. It first extracts the convolutional features of a pre-trained CNN. Then, a Region Proposal Network (RPN) [17] is used to extract the regions of interest. For each region of interest, a shifting and linear layer are used to reduce the dimensionality of the descriptor. The final descriptor is simply a normalized sum of the region-wise descriptors. This network can be learned end-to-end and the obtained descriptor achieves state of the art results, which can be even further improved by using query expansion and database side feature augmentation (TODO either remove or mention before).

Gordo et al [3] found that using very deep networks outperforms the shallower networks. The state of the art results are thus set by a very deep ResNet architecture.

2.3.1 Siamese networks and triplet loss

Figure 2.2 shows the architecture used to train the CNN that produces a descriptor for images, as can be seen in Figure 2.1.

The major difference is that during training, a Siamese architecture is used: the CNN is evaluated with multiple images, using the same weights for each. Then, a combined loss is obtained from the descriptors. Finally, the loss is back-propagated once through all streams of the network, since shared weights are used for all images.

In the case of this particular architecture, three images are evaluated and a triplet loss is used. This triplet loss has previously been used in face recognition tasks by Schroff et al [20]. The triplet loss was first introduced by Weinberger et al [25] as a way of learning the best suited distance metric in a k-nearest neighbor classification problem. This is the reason why the triplet loss was chosen for image retrieval: the goal is to learn a metric which allows to discriminate couples of images containing the same instance from couples of images containing different instances, using the descriptors obtained from the Siamese CNN.

Finally, Gordo et al made further contributions, detailed in their follow-up paper [4].

2.3.2 Dataset

For one, an important step is to use a suitable dataset. Datasets like Oxford5k or Paris6k are too small for a network to learn a distance metric applicable to any type of image. Instead, Gordo et al use the much larger Landmarks dataset, created by Babenko et al [1]. This dataset contains almost 200000 images in total of around 600 different landmarks, mostly buildings.

Another important step is the cleaning of the dataset. The Landmarks dataset contains a lot of noise and weak labels: many images do not represent the building they are labeled with. Gordo et al use a meticulous cleaning process: the main idea is to keep only the largest clusters of matching objects for each of the landmarks. The implementation is based on extracting SIFT and Hessian descriptors, matching them and verifying matches with an affine transformation model. This cleaning process is used in order to annotate the cleaned images with regions of interest as well.

The cleaning requires a lot of processing, but is only done once for the dataset. The cleaned dataset is still fairly large, containing 49000 images and 586 instances of landmarks.

2.3.3 Triplet choice

Another important step is the choice of triplets during training of the Siamese network. Schroff et al [20] and Weinberger et al [25] already identified that using random triplets for training means that most triplets do not contribute to the loss at all, thus leading to vanishing gradients. So it makes sense to try and choose only the hardest triplets: for each reference image, find the image with the same label that is furthest from the reference image according to the current metric (the hardest positive), and the closest from the reference image having a different label (the hardest negative). However, choosing the hardest positive and hardest negative at each step is computationally too expensive. So Schroff et al [20] propose choosing the hardest triplets for each batch. Since there are only a few positive couples available in each batch, we simply choose all positive couples along with the hardest negative for each reference image in every batch. Since this can lead to a collapse in the model, Schroff et al propose choosing semi-hard negatives in the first epochs of training: for each reference image and positive sample, choose the hardest negative that is easier (further according to the metric) than the positive.

The disadvantage of this strategy is that it requires large batches. Instead, Gordo et al [4] uses the following strategy: pre-compute all descriptors in each epoch of training. Then, for each reference image, choose the n easiest positive samples and the m hardest negative samples. Compute the loss for all possible combinations and use the n hardest triplets, with highest loss. This allows using smaller batches, while keeping the cost of computation low, since descriptors are only computed every epoch. It may also help eliminate some noise in the data, since only the easiest positive couples are chosen.

2.3.4 Limitations and goals

The main limitation of Gordo's method is that it aims at producing a general descriptor for comparing any type of image. This means that it is most effective when trained on large datasets, with many samples per instance. For datasets containing images from museums or tourist sites, this is usually not applicable, as there are usually only very few images available for each instance. While there are large datasets, these datasets usually contain a large number of instances as well. This makes it difficult to directly apply the state of the art on our datasets.

On the other hand, it also means that the state of the art in image retrieval wants to avoid over-fit at all costs: the learned descriptor and metric should be able to provide a good metric for any kind of image. While this is usually desired in machine learning, in our case, one goal is to provide a metric specifically designed to discriminate between the images of the given dataset. This is because an audio-guide in a museum can be fine-tuned for that museum, as long as the fine-tuning is computationally cheap enough. So we do not care if a given audio-guide cannot identify images of multiple museums, as long as it can identify images of one museum with high precision.

Another limitation of the state of the art is that it requires a tedious process to clean the dataset and annotate it with regions of interest. This is not suited to museum datasets which are usually very clean to begin with, by the nature of the dataset. Plus, the notion of region of interest is usually not relevant for this type of image: which part of a painting or piece of art contains the semantically relevant information is not obvious.

Finally, an evaluation of the CNN developed by Gordo et al [3] was carried out. We used the published weights obtained in training this CNN on the Landmarks dataset. Figure shows random examples of images in our dataset that were correctly matched with high confidence as well as images that were incorrectly matched. From this evaluation, we can make the following observations:

1. Images that are very similar visually are well matched
2.
3. Images at different scales are not well matched

Thus, a limitation of the current state of the art is matching images, where the object of interest is at different scales.

— 3 —
Contributions

3.1 Fine-tuning a CNN

A first possible approach to this problem is to simply try to fine-tune a classification network to instance classification. This means that we consider each instance as a class, and optimize using a cross-entropy loss, typical for classification.

$$\mathscr{L} = -\frac{1}{N}\sum_{i=1}^{N} y_i \log \hat{y}_i + (1-y_i)\log(1-\hat{y}_i) \tag{3.1}$$

Equation 3.1 describes the cross-entropy loss for a given instance. N is the number of samples, y_i an indicator variable taking the value 1 if sample i belongs to the given instance and 0 otherwise, and $\hat{y}_i$ is the predicted probability that sample i belongs to the given instance.

Fine-tuning a CNN for classification on different data has been studied intensively by Yosinki et al [26]. In particular, their study shows that it is only important to fine-tune the neurons of higher layers of a CNN. Furthermore, they show that fine-tuning can increase generalization even in the fine-tuned model. Both of these properties are desirable in our task, because they reduce the memory and time needed to train a network, as well as the need for a large dataset used for fine-tuning.

The specific considerations we take with regards to fine-tuning are described in the following sections.

3.1.1 Data augmentation

Our datasets contain an average of less than 10 samples per instance. This is too little to train a typical CNN model designed for classification, even when fine-tuning. One way to overcome this is to augment the data, by randomly applying affine transformations, color perturbations and other random transformations.

Since we specifically identified an issue related to different scales in images, it makes sense to augment the data by randomly scaling the images.

We found that randomly rotating and flipping the images improved performance as well.

3.1.2 Transfer learning

The modularity of a CNN means that we can easily transfer the weights from a pre-trained model, and only retrain 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 Proposed approach

— 4 —
Evaluation

4.1 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 [16]. Both datasets are typical of instance search datasets in museums or touristic sites: 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. Table 4.1 lists the content and statistics of the datasets in detail.

4.1.1 CLICIDE

The CLICIDE dataset contains photographs taken in the Grenoble Museum of Art. The objects represented in the photographs are paintings and still images, exclusively. The full dataset contains 207 images with no meaningful content, mostly showing walls. These images are labeled as such and have been filtered from the dataset in our evaluation. This explains the difference between total number of images and number of images in Table 4.1. Furthermore, there are 12 query images which contain objects not present in the reference images. These have been filtered out as well.

The CLICIDE dataset is characteristic because the different images for each instance usually consist of at least one global view of the painting and multiple other images representing sub-regions of the same painting.

4.1.2 GaRoFou

The GaRoFou dataset contains high-resolution photographs taken in the Museun of Fourvière in Lyon. The objects represented in the photographs are a cultural heritage collection: sculptures, steles and small artifacts grouped together in glass cabinets.

The full GaRoFou dataset is larger than the part used here: along with the dataset of images, it contains a dataset of videos as well as images extracted from these videos. The video collection has not been used as part of our research.

	CLICIDE	*GaRoFou*
Type of objects	Paintings	Glass cabinets containing various objects
#Instances	464	311
#Reference images	3245	1068
#Query images (#instances)	165 (134)	184 (166)
Median #images per instance	6	2
Min #images per instance	1	1
Max #images per instance	22	45
Total reference images (#instances)	3452 (473)	1068 (311)
Total query images (#instances)	177 (143)	184 (166)

Table 4.1: Details of the datasets used for evaluation

4.2 Metrics

There are several commonly used metrics in classification and information retrieval. The following sections define these metrics and their use in our research.

4.2.1 Precision

$$P = \frac{TP}{PRED} \tag{4.1}$$

$$P_q = \frac{RR_q}{RET} \tag{4.2}$$

$$P_q@k = \frac{RR_q^k}{k} \tag{4.3}$$

$$MP@1 = \frac{1}{Q}\sum_q P_q@1 \tag{4.4}$$

In classification, precision is a measure of how many correct predictions were made out of all predictions. Equation 4.1 shows this relationship: TP are the true positives, the correct predictions, and $PRED$ is the number of all predictions, including incorrect ones. $PRED$ is usually equal to the size of the testing dataset in classification.

In information retrieval, similarly, precision and more generally Precision@k is a measure of how many relevant documents were retrieved out of all retrieved documents. However, the metric is evaluated for each query, instead of the whole testing dataset. Equations 4.2 and 4.3 illustrate this: P_q is the precision for query q, RR_q represents the number of retrieved and relevant documents for query q. RET represents the total number of retrieved documents. Finally, RR_q^k represents the number of retrieved, relevant documents out of the k first retrieved documents for query q.

Thus, Precision@k simply uses a cut-off at k: it measures how many relevant documents there are in the first k retrieved documents.

In instance search, only the first retrieved document is important: If the first result is correct, we correctly identify the instance. Thus, we are mostly interested in Precision@1 in our research. In order to generalize this metric to all queries, the mean Precision@1 is used, as defined in Equation 4.4. Here, Q represents the number of queries.

4.2.2 Mean average precision

$$R_q = \frac{RR_q}{REL_q} \tag{4.5}$$

$$AP_q = \frac{1}{REL_q} \sum_{k=1}^{RET} rel(RET_k) P@k \tag{4.6}$$

$$MAP = \frac{1}{Q} \sum_{q} AP_q \tag{4.7}$$

In information retrieval, average precision is commonly used, since it allows to consider the order in which documents are returned by a system. Average precision represents the area under the precision-recall curve for a given query q. Recall for query q is defined as in Equation 4.5 as the fraction of retrieved, relevant documents for query q as compared to all relevant documents to query q.

Equation 4.6 defines average precision for a given query q: it can be expressed as a finite sum over the Precision@k values at every possible cut-off k, multiplied by an indicator $rel(RET_k)$, which is 1 if the k-th retrieved document is relevant. This sum is then normalized by the total number of relevant documents for query q REL_q.

Finally, mean average precision MAP represents the mean of all average precision values over Q queries, as defined in Equation 4.7.

In instance search, mean average precision is not important as we only care about the first ranked result. Nevertheless, it can be an interesting metric to allow better comparison between different systems, so we compute the mean average precision for all results along with the mean Precision@1.

4.3 Baselines

Table shows the baselines set for the two evaluation datasets, as described in Section 4.1.

4.4 Performance

4.4.1 Training

4.4.2 Evaluation

— 5 —

Results

— 6 —
Conclusion

6.1 Conclusions from our research

6.2 Future work

— A —

Appendix

Bibliography

[1] Artem Babenko, Anton Slesarev, Alexandr Chigorin, and Victor Lempitsky. Neural codes for image retrieval. In *European conference on computer vision*, pages 584–599. Springer, 2014.

[2] Yoav Freund and Robert E. Schapire. A desicion-theoretic generalization of on-line learning and an application to boosting. In *European conference on computational learning theory*, pages 23–37. Springer, 1995.

[3] Albert Gordo, Jon Almazán, Jerome Revaud, and Diane Larlus. Deep Image Retrieval: Learning Global Representations for Image Search. In *Computer Vision – ECCV 2016*, pages 241–257. Springer, Cham, October 2016.

[4] Albert Gordo, Jon Almazan, Jerome Revaud, and Diane Larlus. End-to-end learning of deep visual representations for image retrieval. *arXiv preprint arXiv:1610.07940*, 2016.

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. *arXiv:1512.03385 [cs]*, December 2015. arXiv: 1512.03385.

[6] Geoffrey E. Hinton, Simon Osindero, and Yee-Whye Teh. A fast learning algorithm for deep belief nets. *Neural computation*, 18(7):1527–1554, 2006.

[7] Geoffrey E. Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R. Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*, 2012.

[8] Gao Huang, Zhuang Liu, Kilian Q. Weinberger, and Laurens van der Maaten. Densely Connected Convolutional Networks. *arXiv:1608.06993 [cs]*, August 2016. arXiv: 1608.06993.

[9] Hervé Jégou, Matthijs Douze, and Cordelia Schmid. Hamming embedding and weak geometry consistency for large scale image search-extended version. 2008.

[10] Alex Krizhevsky, Ilya Sutskever, and Geoffrey 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.

[11] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

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

[13] Andrej Mikulik, Michal Perdoch, Ondřej Chum, and Jiří Matas. Learning Vocabularies over a Fine Quantization. *International Journal of Computer Vision*, 103(1):163–175, May 2013.

[14] James Philbin, Ondrej Chum, Michael Isard, Josef Sivic, and Andrew 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.

[15] James Philbin, Ondrej Chum, Michael Isard, Josef Sivic, and Andrew 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.

[16] Maxime Portaz, Johann Poignant, Mateusz Budnik, Philippe Mulhem, Jean-Pierre Chevallet, and Lorraine Goeuriot. Construction et évaluation d'un corpus pour la recherche d'instances d'images muséales.

[17] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[18] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision*, 115(3):211–252, December 2015.

[19] Bernhard Scholkopf, Kah-Kay Sung, Christopher JC Burges, Federico Girosi, Partha Niyogi, Tomaso Poggio, and Vladimir Vapnik. Comparing support vector machines with Gaussian kernels to radial basis function classifiers. *IEEE transactions on Signal Processing*, 45(11):2758–2765, 1997.

[20] Florian Schroff, Dmitry Kalenichenko, and James 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.

[21] John Shawe-Taylor and Nello Cristianini. *Kernel methods for pattern analysis*. Cambridge university press, 2004.

[22] Karen Simonyan and Andrew Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv:1409.1556 [cs]*, September 2014. arXiv: 1409.1556.

[23] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alex Alemi. Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning. *arXiv:1602.07261 [cs]*, February 2016. arXiv: 1602.07261.

[24] Andrea Vedaldi and Andrew Zisserman. Efficient additive kernels via explicit feature maps. *IEEE transactions on pattern analysis and machine intelligence*, 34(3):480–492, 2012.

[25] Kilian Q. Weinberger, John Blitzer, and Lawrence Saul. Distance metric learning for large margin nearest neighbor classification. *Advances in neural information processing systems*, 18:1473, 2006.

[26] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? *arXiv:1411.1792 [cs]*, November 2014. arXiv: 1411.1792.