CNN FEATURES ARE ALSO GREAT AT UNSUPERVISED CLASSIFICATION

Joris Guérin, Olivier Gibaru, Stéphane Thiery and Eric Nyiri

Laboratoire des Sciences de l'Information et des Systèmes (CNRS UMR 7296)
Arts et Métiers ParisTech, Lille, France

ABSTRACT

This paper aims at providing insight on the transferability of deep CNN features to unsupervised problems. We study the impact of different pretrained CNN feature extractors on the problem of image set clustering for object classification as well as fine-grained classification. We propose a rather straightforward pipeline combining deep-feature extraction using a CNN pretrained on ImageNet and a classic clustering algorithm to classify sets of images. This approach is compared to state-of-the-art algorithms in image-clustering and provides better results. These results strengthen the belief that supervised training of deep CNN on large datasets, with a large variability of classes, extracts better features than most carefully designed engineering approaches, even for unsupervised tasks. We also validate our approach on a robotic application, consisting in sorting and storing objects smartly based on clustering.

KEYWORDS

Transfer learning, Image clustering, Robotics application

1. INTRODUCTION

In a close future, it is likely to see industrial robots performing tasks requiring to make complex decisions. In this perspective, we have developed an automatic sorting and storing application (see section 1.1.2) consisting in clustering images based on semantic content and storing the objects in boxes accordingly using a serial robot (`https://youtu.be/NpZIwY3H-gE`). This application can have various uses in shopfloors (workspaces can be organized before the workday, unsorted goods can be sorted before packaging, ...), which motivated this study of image-set clustering.

As of today, deep convolutional neural networks (CNN) [1] are the method of choice for supervised image classification. Since [2] demonstrated astounding results on ImageNet, all other methods have rapidly been abandoned for ILSVRC [3]. As suggested by [4], performances of CNN are highly correlated to the amount of labeled training data available. Nevertheless, even when few labels are available, several recent papers [5, 6, 7] have shown that CNN can still outperform any other approach by transferring knowledge learned on large datasets. In particular, [5] has shown that extracting features from OverFeat [8] pretrained on ImageNet, and training a simple Support Vector Machine (SVM) [9] classifier on these features to fit the new dataset

provides better results than many more complex approaches for supervised classification. These results demonstrate that a CNN, trained on a large and versatile dataset, learns information about object characteristics that is generic enough to transfer to objects that are not in the original dataset.

While developing the automatic robotic sorting and storing application, we needed to classify sets of images based on their content, in an unsupervised way. Multiple papers introduced methods to solve unsupervised object classification from sets of images (see section 1.1.1), producing relatively good results. However, we wanted to know if the information from a large and versatile dataset, stored in the weights of a CNN, could be used straightforwardly to outperform state-of-the-art algorithms at unsupervised imageset classification. The goal of this paper is to answer the following question: *How good are features extracted with a CNN pretrained on a large dataset, for unsupervised image classification tasks?* To do so, we use a similar approach to [5], consisting in applying classic clustering methods to features extracted from the final layers of a CNN (see section 2 for more details) and comparing it with state-of-the-art image set clustering algorithms [10, 11] on several public datasets.

The intuition behind such approach for unsupervised object classification is that, as it works with SVM [5], the CNN must project data in a feature space where they are somehow linearly separable. Thus, simple clustering algorithms such as K-means might be working well. However, this study is interesting as the performance of such simple clustering algorithms often depends on the notion of distance between points, on which we remain uncertain.

1.1. Previous work

1.1.1. Image-set clustering

Given a set of unlabeled images, the image-set clustering problem consists in finding subsets of images based on their content: two images representing the same object should be clustered together and separated from images representing other objects. Figure 1 illustrates the expected output from an image-set clustering algorithm in the case of our robotics application. This problem should not be confused with image segmentation [12], which is also sometimes called image clustering.

Image-set clustering has been widely studied for two decades. It has applications for searching large image database [13, 14, 15], concept discovery in images [16], storyline reconstruction [17], medical images classification [18], ... The first successful methods focused on feature selection and used sophisticated algorithms to deal with complex features. For instance, [19] represents images by Gaussian Mixture Models fitted to the pixels and clusters the set using the Information Bottleneck (IB) method [20]. [21] uses features resulting from image joint segmentation and sequential IB for clustering. [11] uses Bags of Features with local representations (SIFT, SURF, ...) and defines commonality measures used for agglomerative clustering. Recently, image-set clustering algorithms have shifted towards using deep features. [10] uses deep auto-encoders combined with ensemble clustering to generate feature representations suitable for clustering. [22, 18] learns jointly the clusters and the representation using alternating optimization [23].

1.1.2. Robotic automatic sorting application

Together with showing that deep *features + simple clustering* outperforms other approaches on unsupervised object classification, we apply this pipeline to solve an automatic smart robot sorting application first introduced in [24]. The idea is that the robot, equipped with a camera, visualizes a set of objects placed on a table. Given the number of storage boxes available, it needs to figure out the best way of sorting and storing the objects before physically doing it. The approach proposed in this paper exploits semantic information (deep features) while [24] uses a computer vision algorithm to extract shapes and colors. A video of the application can be seen at (`https://youtu.be/NpZIwY3H-gE`). An example of inputs/output of the application is shown in Figure 1. The robustness of this application is also investigated in section 4 by changing the lighting conditions, the position and orientation of the objects as well as the background. For this robustness validation, we built a dataset that appears to be a challenging one for image-set clustering (`http://www.usine-agile.fr/datas/22-cnn-datas-1.html`).
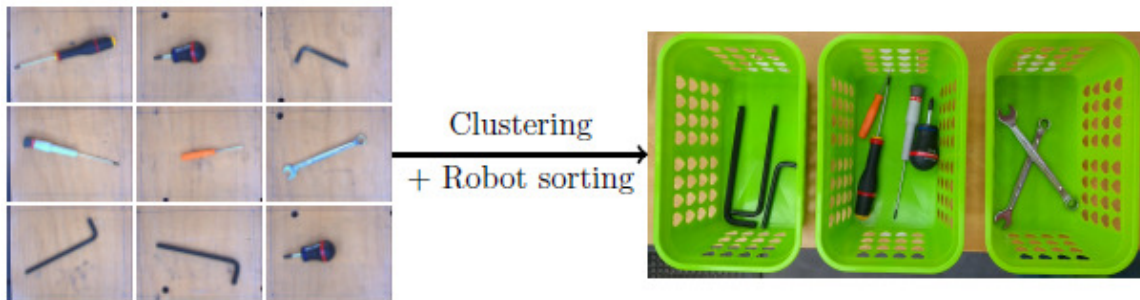

Figure 1: Robotic application description

1.2. Contributions

The main contribution of this paper is to convey further insight into deep CNN features by showing their scalability to unsupervised classification problems. We also propose a new baseline on several image clustering tasks.

Other contributions include the implementation of an application combining unsupervised image classification with robotic sorting. The method proposed to solve this problem, constitutes a new step towards autonomous decision-making robots. The dataset introduced in this paper, which is relatively challenging for image-set clustering, is also a contribution that can be used in further research to investigate robustness to background and lighting conditions for image clustering algorithms.

2. CLUSTERING IMAGES WITH DEEP FEATURE EXTRACTION

2.1. Pipeline description

The pipeline we propose for image set clustering is fairly straightforward. It consists in extracting deep features from all the images in the set, by using a deep convolutional neural network pretrained on a large dataset for image classification and then apply a "standard" clustering algorithm to these features. We initially tried this approach as a first step towards developing a

better clustering algorithm, however, it appears that this simple approach outperforms state-of-the-art algorithm at image-set clustering.

To implement this unsupervised image classification pipeline, we first need to answer four questions:

- What dataset should be used for pretraining?
- What CNN architecture should we use?
- Which layer output should be chosen for feature extraction?
- What clustering algorithm should be used?

As of today, ImageNet is the only very large labelled public dataset which has enough variety in its classes to be a good feature extractor for a variety of tasks. Moreover, there are plenty of CNN pretrained on ImageNet already available online. Hence, we will use a CNN pretrained on ImageNet. The three other questions are answered experimentally. We use the VOC2007 [25] test set without labels, which is new for the pretrained net, to compare performances of the different options.

To ease development and accelerate implementation, we compare the Keras [26] implementations of ResNet50 [27], InceptionV3 [28], VGG16, VGG19 [29] and Xception [30] with the pretrained weights provided by Keras. For the clustering algorithms, we use the scikit-learn [31] implementations of K-means (KM) [32], Minibatch K-means (MBKM) [33], Affinity Propagation (AP) [34], Mean Shift (MS) [35], Agglomerative Hierarchical Clustering (AC) [36], DBScan (DBS) [37] and Birch (Bi) [38]. For each CNN, the layers after which the features are extracted can be found in Table 1 (Layers names are the same as in the Keras implementations).

In the image-set clustering problem, the expected classes are represented by objects present on the picture and for this reason we need semantic information, which is present in the final layers of the network. Thus, we only choose layers among the last layers of the networks. On the one hand, the last one or two layers might provide better results as their goal is to separate the data (at least for the fully-connected layers). On the other hand, the opposite intuition is also relevant as we can imagine that these layers are too specialized to be transferable. These two contradictory arguments motivated the following experiment.

We also note that the test set of VOC2007 has been modified for this validation. We removed all the images presenting two or more labels in order to have ground truth to compute Normalized Mutual Information (NMI) scores. Indeed, if an image possesses several labels we cannot judge if the clustering pipeline classified it properly or not. We note VOC2007-SL (single label) the modified VOC2007 test set.

2.2. Hyperparameters Choice

To answer the questions stated above, we try to cluster the VOC2007-SL set using all combinations of CNN architectures, layer choices and clustering algorithms. To compare performances, we use NMI scores. We also report clustering time for completeness. Only scikit-learn default hyperparameters of the different clustering algorithms are used, which illustrate the simplicity of this approach. For KM and MBKM, as the results depend on random initialization, experiments are run ten times and reported results are averaged over the different runs.

Looking at the results, we choose Agglomerative Clustering on features extracted from the final layer of an Xception CNN pretrained on ImageNet for image-set clustering. This pipeline is then compared to state-of-the-art methods in the next section in order to see how transferable CNN ImageNet features are for unsupervised categorization.

3. VALIDATION ON SEVERAL PUBLIC DATASETS

3.1. Datasets Description

The efficiency of the proposed method is demonstrated by comparing it to other recent successful methods (see section 3.2) on several public datasets which characteristics are described in Table 2.

Table 1: NMI scores (in black) and time in seconds (in blue, italics) on Pascal VOC2007-SL test set using different CNN, different output layers and different clustering algorithms. (Layers names are the same as in the Keras implementations).

		KM	MBKM	AP	MS	AC	DBS [1]	Bi
Inception V3	mixed9	0.108 *374*	0.105 *7.3*	0.219 *12.7*	0.153 *16281*	0.110 *525*	0 *138*	0.110 *577*
	mixed10	0.468 *609*	0.401 *5.1*	0.442 *8.5*	0.039 *12126*	0.595 *525*	0 *119*	0.595 *567*
	avg pool	0.674 *6.3*	0.661 *0.2*	0.621 *7.7*	0.024 *230*	0.686 *8.5*	0 *1.8*	0.686 *8.9*
Resnet 50	avg pool	0.6748 *7.0*	0.641 *0.1*	0.587 *4.6*	0.043 *197*	0.640 *8.0*	0 *1.9*	0.640 *8.9*
VGG 16	block4 pool	0.218 *278*	0.085 *3.6*	0.133 *6.0*	0.124 *10010*	0.277 *391*	0 *82.8*	0.277 *436*
	block5 pool	0.488 *78*	0.048 *1.1*	0.262 *9.3*	0.194 *2325*	0.530 *99*	0 *21*	0.530 *107*
	fc1	0.606 *17*	0.458 *0.2*	0.421 *4.8*	0.187 *365*	0.617 *17*	0 *3.8*	0.617 *19*
	fc2	0.661 *16*	0.611 *0.2*	0.551 *4.3*	0.085 *373*	0.673 *15.9*	0 *3.8*	0.673 *19.7*
VGG 19	block4 pool	0.203 *220*	0.139 *3.7*	0.124 *6.3*	0.135 *10298*	0.234 *388*	0 *83*	0.234 *435*
	block5 pool	0.522 *74*	0.321 *0.9*	0.250 *9.3*	0.198 *2353*	0.540 *97*	0 *20*	0.540 *106*
	fc1	0.607 *17*	0.471 *0.2*	0.449 *9.3*	0.188 *365*	0.628 *17*	0 *3.9*	0.628 *18*
	fc2	0.672 *15*	0.615 *0.2*	0.557 *5.6*	0.083 *391*	0.671 *17*	0 *3.9*	0.671 *18*
Xception	block13 pool	0.376 *410*	0.264 *4.8*	0.351 *10*	0.044 *9677*	0.473 *403*	0 *87*	0.473 *444*
	block14 act	0.574 *935*	0.428 *10*	0.584 *10*	0.071 *24809*	0.634 *820*	0 *180*	0.634 *901*
	avg pool	0.692 *7.1*	0.636 *0.1*	0.636 *4.9*	0.052 *201*	**0.726** ***8.5***	0 *5.5*	0.726 *9.1*

Table 2: Several key features about the datasets used for method validation.

	Problem type	Image Size	# Classes	# Instances
COIL100 [39]	Object recognition	128×128	100	7201
Nisters [40]	Object recognition	640×480	2550	10200
ORL [41]	Face recognition	92×112	40	400
VOC2007-5-ML [2]	Object recognition	variable	5	3376

The clustering tasks involved by these datasets are different from each others (Face recognition, grouping different objects, recognizing different pictures of the same object). In addition, the content of the classes differs from the ones in ImageNet. For these reasons, the four datasets constitute a good benchmark to quantify the robustness of transfer learning for unsupervised object categorization.

3.2. Results Comparison

We propose a comparison with the results reported in the following papers dealing with image set clustering:

[11] proposes different clustering algorithms applied on bags of features. In Table 3, we note "BoF" the best results obtained by such pipeline on the different datasets.

[10] proposes a method called infinite ensemble clustering (IEC). In the paper, IEC algorithm is compared to several other deep clustering algorithms and ensemble clustering algorithms. In Table 3, we report the best results obtained using Deep Clustering (DC) and Ensemble Clustering (EC) for each datasets. We note that for VOC2007-5-ML, [10] also uses deep features as clustering inputs (the CNN used is not reported).

[22] proposes a method called Joint Unsupervised Learning (JULE) of Deep Representations and Image Clusters, based on Alternating optimization between clustering and weight optimization of the CNN feature extractor. Results from this work are reported in Table 3.

For each dataset groundtruth is known as they are intended for supervised classification. We compute both NMI scores and purity for each dataset/method pair.

Table 3 shows that features extracted by the final layer of Xception combined with Agglomerative Clustering performs better than or close to state-of-the-art methods at unsupervised object classification as well as fine-grained image clustering (ORL). Results on the ORL dataset are interesting as they show that pretrained Xception is able to classify different faces without supervision, although ImageNet does not deal with human faces at all.

----

[1]The poor results with DB-Scan might come from the default parameters. We might get better results using different configurations, but this is out of the scope of this paper.

[2]The data used for VOC2007 in [10] are irrelevant for clustering with hard assignment. The VOC2007 subset used in [10] contains images with several objects but only one label. However, we still ran our clustering method on this data to be able to compare results. This second modified VOC2007 set is denoted VOC2007-5-ML (5 classes - multiple labels)

Table 3: NMI scores and purity comparison on various public datasets. (A result that is not reported in the papers cited above is denoted N.R.)

NMI scores					
	DC	EC	BoF	JULE	Ours (Xception + AC)
COIL100	0.779	0.787	N.R.	0.985	0.951
Nisters	N.R.	N.R.	0.918	N.R.	0.991
ORL	0.777	0.805	0.878	N.R.	0.93
VOC2007-5-ML	0.265	0.272	N.R.	N.R.	0.367

Purity			
	DC	EC	Ours (Xception + AC)
COIL100	0.535	0.546	0.882
Nisters	N.R.	N.R.	0.988
ORL	0.578	0.630	0.875
VOC2007-5-ML	0.513	0.536	0.622

This is an important result as it shows that, *with today's methods, given an unlabeled image-set, we can extract more information from a large labeled dataset, with a large variety of classes, than from the set itself.* It is better than hand-engineered approaches (BoF) as well as unsupervised trained deep networks (Deep clustering and Ensemble clustering). It also raises the question of how the representation learning problem should be handled. Indeed, although less satisfactory from a research perspective, it might be more appropriate to work on the creation of a larger database and train networks on it so that it can be used as a knowledge base for many other problems.

We underline the very good results of JULE ([22]) at clustering COIL100. Regarding the results of this methodology on Scene clustering [18], it appears that fine tuning feature extraction using alternating optimization is a good way of improving clustering results. However, the simple approach proposed here still keeps the advantage of being very fast (as it only requires to evaluate the network once for each sample and apply AC), which is useful for our application for instance.

It is also interesting to notice that [10] is also using CNN features for clustering VOC2007-5-ML. The fact that our pipeline outperform largely DC and EC for this dataset illustrate that when using transfer learning for a new problem, careful selection of the CNN used is important, which is often neglected.

3.3. A Word on Scene Clustering

The problem studied in this paper is the one raised by the robotic application, unsupervised objects sorting. For this reason, we compared our result at different object categorization tasks as well as fine-grained classification for clustering of similar objects. Another interesting image classification problem is the one of scene clustering, studied in [18] on two datasets ([42, 43]). For this task, the pipeline proposed in this paper cannot perform as well as for object classification. Indeed, ImageNet does not contain any class which requires to group objects together. Thus, although the features are good at supervised scene classification, without further training they are not able to group objects together as such behaviour is not encoded in the final layers of the CNN. However, this issue is not inherent to the method defined and we believe that with bigger and more versatile datasets, results would be as good as any other method.

4. APPLICATION VALIDATION

In the setting tested initially (Figure 1), where the set of objects to cluster is composed of screw drivers, at keys, allen keys and clamps, and where the background is a table, the success rate of the application described in section 1.1.2 is 100%. Although for certain classes (flat and allen keys) the intra-cluster similarity is high, it is not the case for the two others. This task is also difficult to solve because we carried out the experiments in a shopfloor under unmastered lighting.

For further testing of the application robustness, we have built a dataset for pixel-based object clustering. The full dataset, together with its description, can be found at (http//www.usine-agile.fr/datas/22-cnn-datas-1.html) and example images can be seen on Figure 2. The dataset statistics are reported in table 4. This dataset is difficult because some classes have low intra-cluster similarity (usb) and extra-cluster similarity between some classes is relatively high (pens/screws). The lighting conditions as well as the background also change within the dataset, which makes the task even harder.

Table 4: Several key features about the constructed dataset.

Problem type	Image Size	# Classes	# Instances
Object recognition	640×480	7	560

For each conditions, we randomly pick one out of the several pictures (different position /orientation) of each object. The results reported in the top subtable of Table 5 are averaged over 100 random combinations. The clustering results are not perfect, looking at the misclassifications, the main source of error comes from classes with low intra-cluster similarity (pens, usb) and from background containing sharp edges (conditions 4 and 5).

Figure 2: Example images from the robustness validation dataset. Objects in the same row are expected to be clustered in the same group. The five different background/lighting conditions are represented in this figure.

We also carry out an experiment of fine grained classification with this dataset. Within each class, we try to group together the pictures representing the exact same object. Purity results can be found in the bottom subtable of Table 5. An interesting fact can be noticed about object

recognition within a category. Classes responsible for decreasing the clustering quality in the first subtable are objects with the highest purity in the second table. Such remark makes sens as low intra-cluster similarity is good for the second task but harmful for the first one.

Looking at the results, this robustness validation dataset appears to be a challenging one for image-set clustering and could be used to validate further research in the field.

Table 5: Clustering accuracies for different background, lighting and orientation conditions on the tool clustering dataset.

Grouping by category

Metric \ Condition	1	2	3	4	5	Mixed
Purity	0.85	0.94	0.84	0.69	0.79	0.58
NMI score	0.87	0.94	0.83	0.71	0.82	0.54

Recognizing object inside a category

Object \ Condition	1	2	3	4	5
Allen	0.58	0.58	0.67	0.67	0.67
Clamp	0.83	0.92	0.67	0.58	0.67
Driver	0.75	0.75	0.75	0.63	0.75
Flat	0.58	0.83	0.67	0.58	0.58
Pen	1.0	1.0	0.75	0.69	1.0
Screws	0.5	0.56	0.75	0.50	0.56
USB	0.80	0.65	0.85	0.55	0.65

5. CONCLUSION AND PERSPECTIVES

5.1. Conclusive Remarks

This paper extends the interesting work of [5, 6, 7] about the transferability of CNN features. It shows that, even for unsupervised classification tasks, features extracted from deep CNN trained on large and diverse datasets, combined with classic clustering algorithms, can compete with more sophisticated and tuned image-set clustering methods. The fairly simple and naive pipeline proposed outperforms the best results reported in recent work, which raises the question of which research direction should be chosen to reach generic knowledge. Are efforts spent in developing image representation extractors more useful than simply building larger and more diverse datasets?

This approach is used to implement a robotic application using unsupervised image classification to store objects smartly. To validate this application, we also built a challenging dataset for image clustering that is made available to the research community.

5.2. Future Work

The proposed improvements mainly go in the direction of the robotics application, which is still not robust enough to adapt perfectly to very different looking objects within a cluster and to difficult backgrounds and lighting conditions. If we want to make it work in difficult

environments, the clustering pipeline needs to be improved. One possible direction is to tune the final clustering algorithm, indeed, the scikit clustering algorithms are used without any parameter tuning, setting hyperparameters to their default values.

The sorting application can also be improved by introducing automatic image segmentation, which would make it more suitable for practical uses. To do this, we could use a pretrained region proposal network [44] and cluster objects in the proposed regions.

ACKNOWLEDGEMENTS

REFERENCES

[1]   Y. LeCun, Y. Bengio, and G. Hinton, \Deep learning," Nature, vol. 521, no. 7553,pp. 436{444, 2015.

[2]   A. Krizhevsky, I. Sutskever, and G. E. Hinton, \Imagenet classification with deep convolutional neural networks," in Advances in neural information processing systems, 2012, pp. 1097{1105.

[3]   O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein et al., \Imagenet large scale visual recognition challenge," International Journal of Computer Vision, vol. 115, no. 3, pp. 211{252, 2015.

[4]   S. Uchida, S. Ide, B. K. Iwana, and A. Zhu, \A further step to perfect accuracy by training cnn with larger data," in Frontiers in Handwriting Recognition (ICFHR), 2016 15th International Conference on. IEEE, 2016, pp. 405{410.

[5]   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.

[6]   J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, \How transferable are features in deep neural networks?" in Advances in neural information processing systems, 2014, pp. 3320{3328.

[7]   J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell, \Decaf: A deep convolutional activation feature for generic visual recognition." in Icml, vol. 32, 2014, pp. 647{655.

[8]   P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, \Over-feat: Integrated recognition, localization and detection using convolutional networks," arXiv preprint arXiv:1312.6229, 2013.

[9]   S. Theodoridis and K. Koutroumbas, \Pattern recognition," 2003.

[10]   H. Liu, M. Shao, S. Li, and Y. Fu, \Infinite ensemble for image clustering," in Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016.

[11] T. Fukui and T. Wada, \Commonality preserving image-set clustering based on diverse density," in International Symposium on Visual Computing. Springer, 2014, pp. 258{269.

[12] N. Dhanachandra and Y. J. Chanu, \A survey on image segmentation methods using clustering techniques," European Journal of Engineering Research and Science, vol. 2, no. 1, 2017.

[13] M. Flickner, H. Sawhney, W. Niblack, J. Ashley, Q. Huang, B. Dom, M. Gorkani, J. Hafner, D. Lee, D. Petkovic et al., \Query by image and video content: The qbic system," computer, vol. 28, no. 9, pp. 23{32, 1995.

[14] Y. Gong, M. Pawlowski, F. Yang, L. Brandy, L. Bourdev, and R. Fergus, \Web scale photo hash clustering on a single machine," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 19{27.

[15] Y. Avrithis, Y. Kalantidis, E. Anagnostopoulos, and I. Z. Emiris, \Web-scale image clustering revisited," in Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 1502{1510.

[16] Y. J. Lee and K. Grauman, \Shape discovery from unlabeled image collections," in Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on. IEEE, 2009, pp. 2254{2261.

[17] G. Kim, L. Sigal, and E. P. Xing, \Joint summarization of large-scale collections of web images and videos for storyline reconstruction," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 4225{4232.

[18] X. Wang, L. Lu, H.-c. Shin, L. Kim, M. Bagheri, I. Nogues, J. Yao, and R. M. Summers, \Unsupervised joint mining of deep features and image labels for large-scale radiology image categorization and scene recognition," arXiv preprint arXiv:1701.06599, 2017.

[19] J. Goldberger, S. Gordon, and H. Greenspan, \Unsupervised image-set clustering using an information theoretic framework," IEEE transactions on image processing, vol. 15, no. 2, pp. 449{458, 2006.

[20] N. Tishby, F. C. Pereira, and W. Bialek, \The information bottleneck method," arXiv preprint physics/0004057, 2000.

[21] Y. Seldin, S. Starik, and M. Werman, \Unsupervised clustering of images using their joint segmentation," in Proceedings of the 3rd International Workshop on Statistical and Computational Theories of Vision (SCTV 2003), 2003.

[22] J. Yang, D. Parikh, and D. Batra, \Joint unsupervised learning of deep representations and image clusters," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 5147{5156.

[23] J. C. Bezdek and R. J. Hathaway, \Some notes on alternating optimization," in AFSS International Conference on Fuzzy Systems. Springer, 2002, pp. 288{300.

[24] J. Guerin, O. Gibaru, S. Thiery, and E. Nyiri, \Clustering for different scales of measurement-the gap-ratio weighted k-means algorithm," arXiv preprint arXiv:1703.07625, 2017.

[25] M. Everingham, A. Zisserman, C. K. Williams, L. Van Gool, M. Allan, C. M. Bishop, O. Chapelle, N. Dalal, T. Deselaers, G. Dorko et al., \The pascal visual object classes challenge 2007 (voc2007) results," 2007.

[26] F. Chollet, \Keras," 2015.

[27] 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, 2016, pp. 770{778.

[28] 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.

[29] K. Simonyan and A. Zisserman, \Very deep convolutional networks for large-scale image recognition," arXiv preprint arXiv:1409.1556, 2014.

[30] F. Chollet, \Xception: Deep learning with depthwise separable convolutions," arXiv preprint arXiv:1610.02357, 2016.

[31] L. Buitinck, G. Louppe, M. Blondel, F. Pedregosa, A. Mueller, O. Grisel, V. Niculae, P. Prettenhofer, A. Gramfort, J. Grobler, R. Layton, J. VanderPlas, A. Joly, B. Holt, and G. Varoquaux, \API design for machine learning software: experiences from the scikit-learn project," in ECML PKDD Workshop: Languages for Data Mining and Machine Learning, 2013, pp. 108{122.

[32] D. Arthur and S. Vassilvitskii, \k-means++: The advantages of careful seeding," in Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms. Society for Industrial and Applied Mathematics, 2007, pp. 1027{1035.

[33] D. Sculley, \Web-scale k-means clustering," in Proceedings of the 19th international conference on World wide web. ACM, 2010, pp. 1177{1178.

[34] D. Dueck and B. J. Frey, \Non-metric affinity propagation for unsupervised image categorization," in Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on. IEEE, 2007, pp. 1{8.

[35] D. Comaniciu and P. Meer, \Mean shift: A robust approach toward feature space analysis," IEEE Transactions on pattern analysis and machine intelligence, vol. 24, no. 5, pp. 603{619, 2002.

[36] F. Murtagh, \A survey of recent advances in hierarchical clustering algorithms," The Computer Journal, vol. 26, no. 4, pp. 354{359, 1983.

[37] M. Ester, H.-P. Kriegel, J. Sander, X. Xu et al., \A density-based algorithm for discovering clusters in large spatial databases with noise." in Kdd, vol. 96, no. 34, 1996, pp. 226{231.

[38] T. Zhang, R. Ramakrishnan, and M. Livny, \Birch: an efficient data clustering method for very large databases," in ACM Sigmod Record, vol. 25, no. 2. ACM, 1996, pp. 103{114.

[39] S. Nayar, S. Nene, and H. Murase, \Columbia object image library (coil 100)," Department of Comp. Science, Columbia University, Tech. Rep. CUCS-006-96, 1996.

[40] D. Nister and H. Stewenius, \Scalable recognition with a vocabulary tree," in Computer vision and pattern recognition, 2006 IEEE computer society conference on, vol. 2. Ieee, 2006, pp. 2161{2168.

[41] F. S. Samaria and A. C. Harter, \Parameterisation of a stochastic model for human face identification," in Applications of Computer Vision, 1994., Proceedings of the Second IEEE Workshop on. IEEE, 1994, pp. 138{142.

[42] A. Quattoni and A. Torralba, \Recognizing indoor scenes," in Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on. IEEE, 2009, pp. 413{420.

[43] Z. Xu, D. Tao, Y. Zhang, J. Wu, and A. C. Tsoi, \Architectural style classification using multinomial latent logistic regression," in European Conference on Computer Vision. Springer, 2014, pp. 600{615.

[44] R. B. Girshick, \Fast R-CNN," CoRR, vol. abs/1504.08083, 2015. [Online]. Available: http://arxiv.org/abs/1504.08083

AUTHORS

Joris Guérin received the diplome d'ingénieur (equivalent to M.Sc. degree) from Arts et Metiers ParisTech and the M.Sc. in Industrial Engineering from Texas Tech University, both in 2015. He is currently a Ph.D student at Laboratoire des Sciences de l'Information et des Systémes (LSIS), at Arts et Métiers ParisTech, Lille, France. His current research focuses on Clustering and Reinforcement Learning for Robotics manipulation.

Olivier Gibaru is currently full professor at the Department of Mathematics and Computer Science at ENSAM, Lille campus. He obtained his PhD in applied mathematics in 1997. His main research interests includes: applied mathematics, estimation for robotic applications, geometry, control engineering and high precision mechanical systems. He is the coordinator of the EU Horizon 2020 ColRobot project www.colrobot.eu. He is an active member of the SMAI-SIGMA group which is a national learned society dedicated to Applied Mathematics for the Industrial Applications.

Stéphane Thiery received the Ph.D degree in Automatics from University of Nice-Sophia Antipolis, France, in 2008. He was a post-doctoral fellow in the NON-A team in INRIA-Lille, for eight months in 2009-2010, and joined Arts et Métiers ParisTech Engineering School, as assistant professor in Applied Mathematics and Automatics, in 2010. His current research includes machine learning, real-time parameters estimation, and control of mechanical systems.
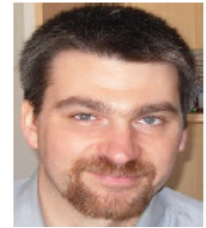
Eric Nyiri received the Ph.D degree in Computer Science from University of Lille I, France, in 1994. He joined Arts et Métiers ParisTech Engineering School, as an assistant professor in Applied Mathematics and Computer Science, in 1995. His initial research domain was L1 interpolation and approximation. In 2010, he joined the LSIS Lab and his current research includes machine learning and path planning for robots. Since 2016, he is a member of the COLROBOT European project.