Discovering and Aligning Discriminative Mid-Level Features for Image Classification

Ronan Sicre and Frédéric Jurie
CNRS UMR 6072 – University of Caen Basse-Normandie – ENSICAEN – France
Email: {ronan.sicre,frederic.jurie}@unicaen.fr

***Abstract*—This paper proposes a new algorithm for image recognition, which consists of (i) modeling categories as a set of distinctive parts that are discovered automatically, (ii) aligning them across images while learning their visual model, and, finally (iii) encode images as sets of part descriptors. The so-obtained parts are free of any appearance constraint and are optimized to allow the distinction between the categories to be recognized. The algorithm starts by extracting a set of random regions from the images of different classes, and, using a *softassign*-like matching algorithm, simultaneously learns the model of each part and assigns image regions to the model's parts. Once the model of the category is trained, it can be used to classify new images by first finding image's regions similar to learned parts and encoding them by the *fisher-on-parts* encoding, which is another contribution of this paper. The proposed framework is experimentally validated on two publicly available datasets, on which state-of-the-art performance is obtained.**

I. Introduction

This paper addresses the task of *image classification*, which consists in predicting whether an image contains an object (or a visual concept) based on the content of the image. This topic has received a lot of attention from the computer vision community since the pioneering work of [1]. The successful approaches rely on the popular bag-of-word model *e.g.* [1], [2], [3] or its variants such as the Fisher vectors [4].

One key issue raised by image classification is how to efficiently use geometric information. While the first works were building on the pure bag-of-words model *e.g.* [1], which consists of pooling the visual features without using their spatial coordinates in any way, it has been shown later (*e.g.* by [3]) that performance can be significantly improved by encoding separately a set of multiple (possibly overlapping) regions, which constitutes a first step toward the use of geometry. Using fixed regions (usually image quad-trees) is obviously limited as the corresponding implicit segmentations of the image is not adapted to the image's content. Several recent works such as [2], [5] have introduced more flexibility by adapting the shape/position of the regions, but the layout was still supposed to be fixed, for a given category.

Observing that images within a category might have very different layouts (*i.e.* spatial organization), it has been shown that categories can be efficiently represented by a set of distinctive regions called *parts* or *fragments* [6], [7], [8], [9]. For example, if 'car' images can be recognized because of the joint presence of 'wheel', 'road' or 'window'-like parts, the position of these regions can be any as long as they are in the image. This idea of introducing some invariance (or alignment) with respect to the position of the parts have been used successfully in the Deformable Part Model of [10]. However, in the case of image classification the relative position of the parts is much less constrained than in the case of object detection. The motivation of our work is precisely to propose a new way to describe images by a set of parts that are aligned across images by construction, without using strong geometric constraints between them. This is achieved by proposing a new model for categories, which states that (i) a category is defined by a set of K parts (ii) these parts are distinctive in the sense that they occur more frequently in the image of the category than in those from other categories (iii) the presence of regions visually similar to the model's part is expected in the image category. These constraints are implemented into an objective function which is optimized during learning. The objective function relies on a *match* function which links model's parts with image regions. Training can be achieved from a set of images describing the category to be recognized, without having to provide any extra annotations. During training, a part classifier is learned simultaneously with the matching of parts and image regions. In a second time, these classifiers can be used to build a visual descriptor of images, also called image signatures: we introduce in this paper the *fisher-on-part* descriptor, consisting in aligning the model on the image and in computing the Fisher vectors of the image regions matched with model parts.

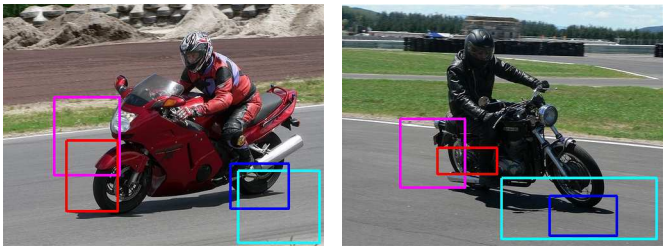

Fig. 1. On this figure, 2 images from the '*riding cycle*' class of the Willow actions dataset are represented with four learned parts

To summarize, the motivation of this paper is to introduce a new framework allowing to automatically discover, learn and align distinctive parts representative of a category, as well as to encode images into a descriptor containing the Fisher vectors of image regions matched with category's parts. The proposed approach is validated on two datasets for which state-of-the-art performance is obtained. The rest of the paper is organized as follows. Related work is presented in section II. Section III provides details on the proposed system that learns, aligns, and encodes distinctive parts. Finally, the experimental validation is given Section IV, before concluding the paper.

II. Related Work

Image classification has been vastly studied in the recent computer vision literature (*e.g.* see the abundant literature related to the Pascal VOC challenge [11]). Most of the modern approaches build on the bag-of-word model [1], following a 4 step pipeline: 1) extraction of local image features, 2) encoding of local image descriptors, 3) pooling of encoded descriptors into a global image descriptor, 4) training and classification of pooled image descriptors for the purpose of object recognition. Several studies evaluate the performance of the first step of the pipeline, namely the low level features *e.g.* gradient, shape, color, and texture descriptors, such as [12], while other propose combining different levels (low - mid - high) of information [13]. Regarding the second step (image encoding) Fisher vectors [4] are considered as achieving state-of-the-art performance at the moment. The (third) pooling step is also shown to provide improvements, and spatial and feature space pooling techniques have been widely investigated [14], [3]. Moreover, [2], [5] have recently proposed two different strategies for embedding spatial information into the bag-of-words framework. Regarding the final step of the pipeline, discriminative classifiers like SVM are widely accepted as the reference in terms of classification performance.

Several authors have shown the importance of adding an intermediate representation [15] – often referred as the mid-level features – for leveraging the performance. We observe three mains trends on mid-level description in the recent literature: hand crafted, learned, and unsupervised features. *Hand crafted mid-level features* aim at encapsulating information on groups of pixel such as superpixels [16], [17], patches [18] or segments [19]. These descriptors are computed similarly for any given image and do not require any learning. On the other hand, a large variety of *learned mid-level features* have been proposed. One of the first one was the Deformable Part Model, proposed by [10]. Improvement have been further achieved by using appearance based clustering and sub-categories [20] and by enforcing steerability and separability of the features [21]. Similarly, semantic attributes [22], [23] have received a lot of interest. Within the learned mid-level features techniques, we observe a large variety in the nature of the learning data. While some feature are based on extra training data such as labeled fragments [24], sketch tokens [25] or pre-trained object detectors [26], most methods use a standard split of training and testing data to learn the distinctive features, as the *structural element patch model* [27] or the *blocks that shout* [8]. Finally, regarding *unsupervised mid-level features*, the work of [28] aims at detecting distinctive patches in an image dataset without any label information.

Our work aims at learning distinctive parts without extra annotations. Therefore, closely related work includes the Deformable Part Model (DPM) [10]. The DPM models categories by using a mixture of parts and classify image regions as object vs non object regions. Classifiers are applied to a representation in which the parts are aligned, by shifting the parts with respect to the root filter. However, for image classification, the variability part positions as well as the variation of appearance within a category makes the problem different.

Our work also bears similarities with [6], which tries to discover the *fragments* that maximize the mutual information between the category and the presence of the fragment in the image. However, [6] suffers from that (i) contrarily to [10], part are just image patches and not discriminative classifiers (ii) the decision is made by verifying the presence of the fragments in the image, instead of training a classifier taking fragment descriptors as input. Our approach takes the advantages of both approaches without having their drawbacks.

More recently, [7] proposes a learning framework for the automatic discovery of image's parts, assuming that partial correspondence between instances of a category are available. These partial correspondences allow the training of part detectors, used in a first time to extract candidates regions. While we share the same motivations, our approach does not require any supervision. In addition, it is worth mentioning [8] and [9] which both propose algorithms for learning parts that are good representatives of a given category. Our work follows the same objectives, without the localization constraints imposed by [9] and the large computation requirement and unoptimized encoding of [8].

This work finally shows the importance of mid-level information and justifies its use to improve recognition capabilities.

III. Proposed method

As explained in the introduction, the proposed method aims at automatically discovering model parts and aligning them with images regions, as well building a description of images based on the so-aligned regions. This section first presents the category model and its associated cost function, which is to be optimized during learning. Secondly, we explain how the parameters of the model can be learned using an iterative framework inspired from the softassign algorithm. Then, more details are given on the algorithm initialization step. Finally, we explain how image signatures can be computed using the learnt model.

A. Category model and objective function

Let us first introduce some notations. We assume having a set of images belonging to the category to be modeled, considered as positive training images and denoted as $\mathcal{I}^+$. $|\mathcal{I}^+|$ represents the number of positive images. In the same way, $\mathcal{I}^-$ is the set of (negative) images belonging to other categories. The whole training set is denoted as $\mathcal{I} = \mathcal{I}^+ \bigcup \mathcal{I}^-$ and contains $|\mathcal{I}|$ images. From each image $I \in \mathcal{I}$, we extract a dense random set of image regions denoted as $\mathcal{R}_I$. Each region r is represented by its signatures x_r, which is, in practice, the bag-of-word representation of the region. The model of the category includes a set of parts denoted as $\mathcal{P}$. The number of parts, $K = |\mathcal{P}|$, is fixed. In the following, $p \in \mathcal{P}$ denotes one of these parts.

The model relies on two assumptions: first, it is expected that each part of the model is present in each positive image. Second, parts should be representatives of the category, which means that they should occur more frequently in positive images than in negative ones.

We implement the first constraint by introducing the match function $m(r, p)$ associating model parts and image regions, and by imposing that $\forall I \in \mathcal{I}^+$ and $\forall p \in \mathcal{P}$, $\sum_{r \in I} m(r, p) = 1$. The match function is defined as:

$$m(r,p) = \begin{cases} 1 & \text{if region } r \text{ is assigned to part model } p \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

In practice, the match function can be seen as a binary matrix with one row per part and one column per image region. We add another constraint ensuring that an image region can be assigned to at most one part, which is written as: $\forall I \in \mathcal{I}^+$, $\forall r \in I$, $\sum_{p \in \mathcal{P}} m(r,p) \leq 1$.

Regarding the second assumption, which states that regions should be discriminative, one way to achieve this would be to measure to which extent each part can be matched with regions from the negative set, and promote those occurring more on positive images. However, such process would be very costly. Therefore, as suggested by [8], we use the LDA technique of [29], which consists in learning once and for all a universal model of negative patches. In practice, the parameter vector w of a part classifier, corresponding to the part p, is defined simply as:

$$w(p,m) = \Sigma^{-1} \left(\frac{\sum_{r \in I, \forall I \in \mathcal{I}^+} m(r,p) \times x_r}{|r \in I, \forall I \in \mathcal{I}^+|} - \frac{\sum_{r \in I, \forall I \in \mathcal{I}} x_r}{|r \in I, \forall I \in \mathcal{I}|} \right) \tag{2}$$

where Σ is the covariance matrix obtained by taking the whole set of regions from both positive and negative images. Consequently, the part models $w(p,m)$ are fully defined once the match function is defined. In addition, the similarity between a region r and a part p of the model can by computed as $w^T(p,m) \times x_r$.

The model is thus fully defined by giving the match function $m(r,p)$. Following the afore mentioned constraints, we define the optimal match function, denoted as $\hat{m}$, as the one maximizing:

$$\begin{cases} \hat{m} = & \arg\max_m \sum_{p \in \mathcal{P}} \sum_{I \in \mathcal{I}^+} \sum_{r \in I} m(r,p) \times w^T(p,m) \times x_r \\ & s.t.\ \forall I \in \mathcal{I}^+,\ \ \forall p \in \mathcal{P},\ \sum_{r \in I} \hat{m}(r,p) = 1 \\ & s.t.\ \forall I \in \mathcal{I}^+,\ \forall r \in I,\ \sum_{p \in \mathcal{P}} m(r,p) \leq 1 \end{cases} \tag{3}$$

Learning this model therefore consists in the (combinatoric) optimization of Eq. (3). Finding the global optimum is not computationally feasible, nevertheless we propose to adapt the point matching algorithm of [30] to obtain an approximate solution, as explained in the following section. This algorithm was first introduced to solves simultaneously the correspondence problem as well as the pose estimation of 3D and 2D data. In [30], two sets of points X_j and Y_k are related by a geometric transformation. Both sets can contain outliers. The *match matrix* m_{jk} is defined as the correspondence matrix such that $m_{jk} = 1$ if point X_j corresponds to point Y_k and 0 otherwise. The problem is further presented as finding the *pose* (*i.e.* the geometric transformation) and the corresponding match matrix m_{jk} that best relates the two sets of points. These two problems are finally solved simultaneously using an iterative process aiming at minimizing an energy function.

B. Learning the match function using softassign

Our main goal is now to efficiently find a good (sub-optimal) solution of the objective function given by Eq. (3). If we ignore, for the moment, the inequality constraint (last constraint of Eq. 3), then the match matrix m can be seen a permutation matrix. We use the *deterministic annealing* method of [31] to turn our combinatoric problem into a continuous one, making the optimization simpler and more efficient. The key idea is to minimize a sequence of objective functions controlled by a parameter β representing the inverse temperature of the system. By increasing the parameter, the objective functions leans towards the discrete function.

The constraints are then relaxed from a permutation matrix constraints to *doubly stochastic matrix* constraints, meaning that every row and column of the matrix should sum up to 1 (see [30] for more explanations). Therefore, the computation of the match function can be achieved iteratively using the *softmax* formulation:

$$\forall I \in \mathcal{I}^+,\ \forall r \in I,\ m(r,p) = \frac{\exp(\beta \times w^T(p,m^*) \times x_r)}{\sum_{r \in I} \exp(\beta \times w^T(p,m^*) \times x_r)} \tag{4}$$

Where $w(p,m^*)^T \times x_r$ is the score function relating the similarity between the part p and the region r of the image I, using the match function m^* computed at the previous iteration. Such a formulation does produce values in the interval $[0,1]$, which is expected. Furthermore, when $\beta \to \infty$, there will be one region per image for which $m(r,p) = 1$, while for the other ones $m(r,p) = 0$, therefore satisfying the first constraint.

In practice, we experimentally observed that the previous formulation tends to favor parts converging to the same 'mean part' for small values of β. Therefore, we utilized the following formulation, which leads to better performance as it encourages sparser representations. $\forall I \in \mathcal{I}^+$ and $\forall r \in I$,

$$m^\dagger(r,p) = \exp\left(\beta\left((w^T(p,m^*) \times x_r) - \max_{\forall r \in I}(w^T(p,m^*) \times x_r)\right)\right) \tag{5}$$

In addition, the match matrix m has to satisfy the doubly stochastic constraints. This can be achieved by using Sinkhorn (see more details in [30]), by iteratively normalizing rows and columns, see Algorithm 1.

Up to this point, we ignored the inequality constraint stating that $\forall I \in \mathcal{I}^+$ and $\forall r \in I$, $\sum_{p \in \mathcal{P}} m(r,p) \leq 1$. We address it by turning the inequality constraint into an equality constraint by adding a slack variable [32], which can be seen as an additional part (the $K+1$-th part of the model) denoted as p_s, with the specificity that this extra part can be matched with several image regions (contrarily to regular parts). Most image regions do not match any part of the model and are therefore matched with this slack part. The constraint becomes: $\forall I \in \mathcal{I}^+$ and $\forall r \in I$, $\sum_{p \in \mathcal{P} \bigcup \{p_s\}} m(r,p) = 1$. This constraint is then satisfied by doing a column normalization of m, which is to say that $\forall I \in \mathcal{I}^+$ and $\forall r \in I$,

$$m(r,p) = \frac{m^\dagger(r,p)}{\sum_{\forall p \in \mathcal{P} \bigcup \{p_s\}} m^\dagger(r,p)} \tag{6}$$

Please note that at the initial values of $m^\dagger(r,p_s)$ are arbitrarily set to $1/K$. The resulting algorithm is Algorithm 1.

C. Initialization of the match function

As the match function m is refined iteratively, initial values of m are required at the beginning. The convergence of the algorithm is shown to be better if parts are initialized to some discriminative regions. To select distinctive initial regions, we first extract the signatures x_r of the regions sampled from positive training images. These signatures are then clustered, using K-means. Here again we use the LDA acceleration of [29], which means that for each cluster, the classifier w is defined as $w = \Sigma^{-1}(\bar{x} - \mu_0)$ where $\bar{x}$ is the average of the signatures within the cluster and μ_0 and Σ the overall mean and covariance matrix.

These classifiers are further applied on the regions of the training images. Maximum responses to the classifiers are then selected per image and averaged over positive and negative subsets, giving us the two scores s^+ and s^-, for a given cluster j, defined as:

$$\begin{array}{l} s_j^+ = \frac{1}{|\mathcal{I}^+|} \sum_{r^* \in \mathcal{I}^+} w_j^T x_{r^*} \\ s_j^- = \frac{1}{|\mathcal{I}^-|} \sum_{r^* \in \mathcal{I}^-} w_j^T x_{r^*} \end{array} \quad (7)$$

Where $\forall I \in \mathcal{I}^+$, $r^* = \arg\max_{r \in I}(w_j^T x_r)$. Then, we denote as C_p the K clusters having the largest s_j^+/s_j^- ratios, which are selected as initial regions. These initial regions are further used to compute the initial part classifier $w(p, m_0)$ as : $w(p) \leftarrow \Sigma^{-1}(C_p - \mu_0)$, used to compute the initial match matrix $m_0(r, p)$.

D. Computing image signatures

Once the model is trained, images can be represented by their signatures. Let us denote as I an image to be encoded. We first extract a set of regions $r \in I$ and compute their corresponding descriptors x_r. We can measure to which extend each region is similar to one of the model parts by using the scoring function defined previously by Eq. (2) as $w^T(p, m) \times x_r$, where m is the match function learned during training.

Then, we want to pool the per part similarities to produce a signature of the image. We propose two different strategies: the bag-of-parts inspired from [8] and a novel approach so-called the *fisher-on-parts*.

Encoding images with bag-of-parts: To compute the bag-of-parts (BOP), the per parts scores are computed for each extracted region on an image. The signature of the image is then given by aggregating, for each part of the model, the average and the maximum of the region scores. Namely, if p_j is one of the K parts of our model, the signature of the image I will be represented by the two following components:

$$\frac{\sum_{r \in I} w^T(p_j, m) \times x_r}{|r \in I_t|} \quad \text{and} \quad \max_{r \in I_t} w^T(p_j, m) \times x_r \quad (8)$$

When the problem is a multi-class problem, we do the same for each class and aggregate the results. Therefore, we obtain a $2 \times K \times C$-dimensional descriptor, where C is the number of classes.

Initialization: $w(p) \leftarrow \Sigma^{-1}(C_p - \mu_0)$
while $\beta \leq \beta_f$ **do**
  while $m^\dagger$ *not converged or # of iteration* $\leq I_0$ **do**
    update match matrix by softassign
    Compute $m^\dagger(r, p)$, based on Eq. 5
    while $\hat{m}^\dagger$ *not converged or # of iteration* $\leq I_1$ **do**
      $\forall I \in \mathcal{I}^+$
      Update $\hat{m}$ by normalizing rows
      $\hat{m}_1^\dagger(r,p) \leftarrow \frac{\hat{m}_0^\dagger(r,p)}{\sum_{r \in I} \hat{m}_0^\dagger(r,p)}$
      Update $\hat{m}$ by normalizing columns
      $\hat{m}_0^\dagger(r,p) \leftarrow \frac{\hat{m}_1^\dagger(r,p)}{\sum_{p \in \mathcal{P} \cup \{p_s\}} \hat{m}_1^\dagger(r,p)}$
    end
    update parts using LDA
    Compute $w(p, m_0^\dagger)$, based on Eq. 2.
  end
  $\beta \leftarrow \beta_r \beta$
end

Algorithm 1: Algorithm for learning the mach function.

Encoding images with Fisher-on-parts: Fisher-on-parts (FOP) aims at encoding together the maximum response of each parts in an image I_t. As in BOP, scores are computed for each region. Then, instead of aggregating average and maximum scores as for the BOP, the maximum scoring region r^* for the part p is selected, as follows:

$$r^* = \arg\max_{r \in I} w^T(p, m) \times x_r \quad (9)$$

Finally, a Fisher vector is computed on the area of the image covered by the K selected regions r^*. Therefore, the final FOP descriptors is $2 \times G \times D \times C$-dimensional vector, where G is the number of Gaussian in the mixture model of the Fisher vector, D is the dimensionality of SIFT descriptors and C the number of categories.

IV. Experiments

A. Datasets

Three classification datasets are utilized to experimentally validate the proposed approach. The Willow dataset [33] aims at classifying 7 human actions in still images. The Boats Datasets goal is to classify several types of boats, while the MIT 67 dataset [34] contains images of 67 types of scenes, to be recognized.

The Willow actions dataset [33] is a database for action classification on unconstrained consumer images from the Internet. The dataset contains 911 images split into 7 classes of common human actions, e.g. 'running', 'riding cycle' etc. There are at least 108 images per actions, with 70 images used as training and the rest as testing images. We note that the dataset also offers bounding boxes fitted on humans performing the actions. In our case, we perform the test without using those bounding boxes, as we want to detect parts automatically without any prior knowledge on the scenes.

The MIT 67 scenes dataset [34] is composed of 67 categories of indoor scenes. These categories include stores (e.g. bakery, toy store), home (e.g. kitchen, bedroom), public spaces

(e.g. library, subway), leisure (e.g. restaurant, concert hall), and work (e.g. hospital, TV studio). Some scenes can be best characterized by their global layout (corridor), or by the objects they contain (bookshop). Each category has around 80 images for training and 20 for testing.

The RECONSURVE Boats Classification Dataset[1] is composed of 2877 images divided in 5 categories of boats (*e.g.* boating, fishing, merchant ship, tanker, passenger).

B. Classification pipeline

Extraction of initial regions: for each image, a set of initial regions is generated by randomly sampling 2,000 regions over the entire image.

Regions descriptors: to obtain region descriptors, dense SIFT points are first extracted on each image using VLFEAT [35] (we use the default 4 scales and sample points every 3 pixels). The SIFT points are further square-rooted to get rootSIFT and the feature dimension is reduced to 80 using PCA [36]. Then each region is characterized using a 1,000-dimensional bag-of-word.

Parameters of the learning algorithm: regarding the learning algorithm, we empirically set the parameters following the choices of [30]: $\beta = 0.41$, $\beta_r = 1.245$, $\beta_f = 1.2$, $I_0 = 4$, $I_1 = 30$ (see Algorithm 1 for the definition of these parameters). The algorithm iterates over the estimation of m until the sum over m of the absolute difference between two iterations is smaller than $\epsilon = 0.005$.

Baseline pipeline: Our approach is compared to the state-of-the-art pipeline [36]. Bag-of-words and Fisher vectors are computed on the root SIFT of the full image. Fisher vectors are also computed using the two first layers (*i.e.* 1×1 and 2×2 segments) of the spatial pyramid.

C. Results

In this section, we first comment on the quantitative results then show some qualitative results (visualization of learned parts) in Figures 1 and 2. In the following, the performance on the three datasets are measured using the mean Average Precision (MAP).

First, we evaluate the impact of the initialization step in the part-learning process, on the Willow dataset. The objective is to measure the contribution of the initialization process described in section III-C over a simple random initialization of the parts. If we randomly initialize the match function we observe a mAP of 46.0% (with the bag-of-parts encoding). Adding the clustering-based initialization improves the mAP by 5% (51.0%). In addition, to prove the usefulness of the proposed algorithm, we evaluated the performance obtained by initializing the match function with the clustering-based approach and without doing the optimization of m. The performance drops to 46.7%, proving that the proposed algorithm improves significantly over discriminative parts learned by clustering.

The bag-of-parts and Fisher-on-parts are then evaluated on the three datasets (see Table I and Table II). For Willow actions, the performance of the two baseline algorithms (bag-of-words and Fisher vectors) are respectively of 50.0% and 58.1%. One can notes that the bag-of-parts slightly outperforms the standard bag-of-word. More interestingly, the propose Fisher-on-parts representation largely outperforms Fisher vectors by more than 3%. Please note that the proposed approach does not use any extra annotations, contrarily to most of the proposed approaches (*e.g.* [9] bounding boxes giving). It explains why we do not provide any comparisons with these methods, as they would be meaningless. The Boats dataset also shows improvement on both the bag-of-parts and the fisher-on-parts. Furthermore, we observe the best performances for the bag-of-parts with 21% and 6% MAP increase over BOW and Fisher vectors respectively. Concerning MIT 67, we first observe that our bag-of-parts encoding offers better performances than the bag-of-word model as well as the bag-of-parts proposed in [8]. We also notice that our Fisher-on-parts improves on the two previous methods. However, we do not obtain better performance than the Fisher vectors extracted on the full image. We believe that this result is due to the fact that the MIT 67 requires a lot of context information to recognize scenes, while our Fisher-on-parts encoding acts as a pooling system that encapsulates most information on the foreground. Combining Fisher-on-parts with Fisher vectors on the whole image (with SPM) gives a MAP of 60.0%, which is significantly better than any other approach.

TABLE I. RESULTS ON WILLOW AND BOATS DATASET. SEE TEXT FOR DETAILS.

Method	Willow (MAP)	Boats (MAP)
bag-of-words [36]	0.500	0.673
Fisher vectors [36]	0.581	0.827
bag-of-parts	0.510	0.888
Fisher-on-parts	0.614	0.837

TABLE II. RESULTS ON MIT 67 SCENES DATASET. SEE TEXT FOR DETAILS.

Method	MAP
bag-of-words [36]	0.345
Fisher vectors [36]	0.550
bag-of-parts of [8]	0.373
our bag-of-parts	0.401
Fisher-on-parts	0.545
Fisher-on-parts based combination	0.600

These experiments show that our descriptors, based on distinctive parts learning, are capable of incorporating mid-level information that can be combined with the full image representation to obtain richer representations.

V. CONCLUSIONS

In this paper, we propose a new algorithm for image recognition by modeling categories as set of distinctive parts that are discovered automatically and aligned across images, while learning their visual model. The parts that are discovered are free of any appearance constraint and allow the distinction between the categories to be recognized. We show how to use the softassign matching algorithm, to simultaneously learn the part models and assign image regions to model's parts, starting from an initial set of randomly extracted image regions. We validated the proposed algorithm on three different datasets on which state-of-the-art performances are obtained.

[1] can be downloaded from https://jurie.users.greyc.fr

Fig. 2. This figure shows the highest scoring regions for a set of parts learned for the *riding horse* action. Each row correspond to a part

ACKNOWLEDGMENT

This work is partly funded by the RECONSURVE project.

REFERENCES

[1] G. Csurka, C. R. Dance, L. Fan, J. Willamowski, and C. Bray, "Visual categorization with bags of keypoints," in *Intl. Workshop on Stat. Learning in Comp. Vision*, 2004.

[2] J. Krapac, J. Verbeek, and F. Jurie, "Modeling spatial layout with Fisher vectors for image categorization," 2011.

[3] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, vol. 2. IEEE, 2006, pp. 2169–2178.

[4] F. Perronnin, J. Sanchez, and T. Mensink, "Improving the fisher kernel for large-scale image classification," in *ECCV*, 2010.

[5] J. Krapac, J. Verbeek, and F. Jurie, "Learning tree-structured descriptor quantizers for image categorization," 2011.

[6] M. Vidal-Naquet and S. Ullman, "Object recognition with informative features and linear classification," in *ICCV*, 2003, pp. 281–288.

[7] S. Maji and G. Shakhnarovich, "Part discovery from partial correspondence," *CVPR*, vol. 0, pp. 931–938, 2013.

[8] M. Juneja, A. Vedaldi, C. V. Jawahar, and A. Zisserman, "Blocks that shout: Distinctive parts for scene classification," in *CVPR*, 2013.

[9] G. Sharma, F. Jurie, C. Schmid *et al.*, "Expanded parts model for human attribute and action recognition in still images," in *CVPR*, 2013.

[10] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *Trans. PAMI*, vol. 32, no. 9, pp. 1627–1645, 2010.

[11] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL Visual Object Classes Challenge 2012 Results."

[12] N. Pinto, Y. Barhomi, D. Cox, and J. DiCarlo, "Comparing state-of-the-art visual features on invariant object recognition tasks," in *Applications of Computer Vision (WACV), 2011 IEEE Workshop on*, 2011.

[13] S. Zheng, Z. Tu, and A. Yuille, "Detecting object boundaries using low-, mid-, and high-level information," in *CVPR*, 2007.

[14] Y. Boureau, N. Le Roux, F. Bach, J. Ponce, and Y. LeCun, "Ask the locals: multi-way local pooling for image recognition," in *ICCV*, 2011.

[15] D. Parikh, "Recognizing jumbled images: the role of local and global information in image classification," in *Computer Vision (ICCV), 2011 IEEE International Conference on*. IEEE, 2011, pp. 519–526.

[16] J. Tighe and S. Lazebnik, "Superparsing: scalable nonparametric image parsing with superpixels," in *ECCV 2010*. Springer, 2010.

[17] R. Sicre, T. E. Tasli, and T. Gevers, "Superpixel based angular differences as a mid-level image descriptor," in *ICPR*, 2013.

[18] B. Fernando, E. Fromont, and T. Tuytelaars, "Effective use of frequent itemset mining for image classification," in *Computer Vision–ECCV 2012*. Springer, 2012, pp. 214–227.

[19] J. Carreira, R. Caseiro, J. Batista, and C. Sminchisescu, "Semantic segmentation with second-order pooling," in *Computer Vision–ECCV 2012*. Springer, 2012, pp. 430–443.

[20] S. K. Divvala, A. A. Efros, and M. Hebert, "How important are deformable parts in the deformable parts model?" in *ECCV 2012. Workshops and Demonstrations*. Springer, 2012, pp. 31–40.

[21] H. Pirsiavash and D. Ramanan, "Steerable part models," in *CVPR*. IEEE, 2012, pp. 3226–3233.

[22] Z. Niu, G. Hua, X. Gao, and Q. Tian, "Context aware topic model for scene recognition," in *CVPR*. IEEE, 2012, pp. 2743–2750.

[23] Y. Su and F. Jurie, "Improving image classification using semantic attributes," *International journal of computer vision*, vol. 100, no. 1, pp. 59–77, 2012.

[24] Z. Liao, A. Farhadi, Y. Wang, I. Endres, and D. Forsyth, "Building a dictionary of image fragments," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, 2012, pp. 3442–3449.

[25] J. J. Lim, C. L. Zitnick, and P. Dollár, "Sketch tokens: A learned mid-level representation for contour and object detection." CVPR, 2013.

[26] I. Endres, K. J. Shih, J. Jiaa, and D. Hoiem, "Learning collections of part models for object recognition." CVPR, 2013.

[27] J. Chua, I. Givoni, R. Adams, and B. Frey, "Learning structural element patch models with hierarchical palettes," in *CVPR 2012*. IEEE, 2012, pp. 2416–2423.

[28] S. Singh, A. Gupta, and A. A. Efros, "Unsupervised discovery of mid-level discriminative patches," in *ECCV*. Springer, 2012, pp. 73–86.

[29] J. M. B. Hariharan and D. Ramanan, "Discriminative decorrelation for clustering and classification," in *ECCV*, 2012.

[30] S. Gold, A. Rangarajan, C.-P. Lu, S. Pappu, and E. Mjolsness, "New algorithms for 2d and 3d point matching:: pose estimation and correspondence," *Pattern Recognition*, vol. 31, no. 8, pp. 1019–1031, 1998.

[31] D. Geiger and F. Girosi, "Parallel and deterministic algorithms from mrfs: Surface reconstruction," *Trans. PAMI*, vol. 13, no. 5, 1991.

[32] V. Chvatal, "Linear programming. 1983."

[33] V. Delaitre, I. Laptev, and J. Sivic, "Recognizing human actions in still images: a study of bag-of-features and part-based representations." in *BMVC*, vol. 2, no. 5, 2010, p. 7.

[34] A. Quattoni and A. Torralba., "Recognizing indoor scenes," 2009.

[35] A. Vevaldi and B. Fulkerson, "Vlfeat an open and portable library of computer vision algorithms," in *ACM Multimedia*, 2010.

[36] K. Chatfield, V. Lempitsky, A. Vedaldi, and A. Zisserman, "The devil is in the details: an evaluation of recent feature encoding methods," 2011.