Discriminative part model for visual recognition

Ronan Sicre, Frédéric Jurie

Discriminative part model for visual recognition

Ronan Sicre and Frédéric Jurie

GREYC, CNRS UMR 6072, University of Caen Basse-Normandie, ENSICAEN, France
Email: {ronan.sicre,frederic.jurie}@unicaen.fr

Abstract

The recent literature on visual recognition and image classification has been mainly focused on Deep Convolutional Neural Networks (Deep CNN) and their variants, which has resulted in a significant progression of the performance of these algorithms. Building on these recent advances, this paper proposes to explicitly add translation and scale invariance to Deep CNN-based local representations, by introducing a new algorithm for image recognition which is modeling image categories as a collection of automatically discovered distinctive parts. These parts are matched across images while learning their visual model and are finally pooled to provide images signatures. The appearance model of the parts is learnt from the training images to allow the distinction between the categories to be recognized. A key ingredient of the approach is a *softassign*-like matching algorithm that simultaneously learns the model of each part and automatically 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 finding image's regions similar to the learned parts and encoding them in a single compact signature. The experimental validation shows that the performance of the proposed approach is better than those of the latest Deep Convolutional Neural Networks approaches, hence providing state-of-the art results on several publicly available datasets.

Keywords:
Computer Vision, Image Classification, Visual Recognition, Part-based

Models.

1. Introduction

The arrival of effective approaches based on Deep Convolutional Neural Networks (Deep CNN), such as the remarkable work of Krizhevsky *et al.* [1] has been perceived as a new trend in image classification, relegating the not so distant approaches such as the bag-of-words [2, 3, 4] or the even more recent Fisher vectors [5] to what some consider now to be a legacy of previous time.

Since then, the literature on *image classification* – the task consists in predicting whether an image contains an object or, more generally, a visual concept based on the content of the image – has benefited from a revival of interest because of the new perspective Deep CNN provides (*e.g.* [6, 7, 8, 7, 9], to cite only a few recent of them).

However, even if Deep CNN obtain very good performance, most of the recent approaches do not explicitly model objects or scenes as deformable configurations which can potentially result in a lack of robustness to appearance/viewpoint changes. One can see this as a limitation, since scenes (and therefore images) can be seen as spatial arrangements of objects or parts, and a decomposition into distinctive parts can results in more expressive and discriminative models [10, 11, 12].

One motivation of this paper is hence to bring together the advantages of Deep CNN and part-based model. The results achieved by Oquab[13] *et al.* constitute one interesting step toward that end. They indeed shown that it is possible to transfer image representations learned with CNNs trained on large datasets to different tasks, even in presence of limited training data. Their method uses ImageNet pre-trained layers of CNN to compute mid-level image signature and can be utilized as an efficient feature encoding system. We use this framework as an alternative to Bag-of-words (BOW) or Fisher vector to encode image regions.

Another key issue raised by the representation of images in the context

of image classification, is how to efficiently use geometric information and, as aforementioned, how to decompose images into stable and distinctive regions. While the early works were building on pure bag-of-words *e.g.* [2], which consists of pooling the visual features without using their spatial coordinates in any way, it has been shown later (*e.g.* by [4]) 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 more recent works such as [3, 14, 15] have introduced more flexibility by adapting the shape/position of the regions, but a strong limitation of these works is that the layout of images is still supposed to be fixed, for a given category.

The proposed work starts with the observation that images within a given category can have very different layouts or spatial organization, even if they can be interpreted globally as sharing the same meaning. In line with this observation, several recent works have shown that categories can be efficiently represented by a set of distinctive regions either called *parts* or *fragments* [16, 10, 11, 12], see Figure 1. 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 successfully utilized in the Deformable Part Model of [17]. However, in the case of image classification the relative position of the parts is much less constrained than in the case of object detection.

In reaction to these observations and concerns, another 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 having to use strong geometric constraints between them. This is achieved by proposing a new model for categories, which is based on the fact that (i) a category is defined by a set of K parts (ii) these parts are distinctive in the sense that they

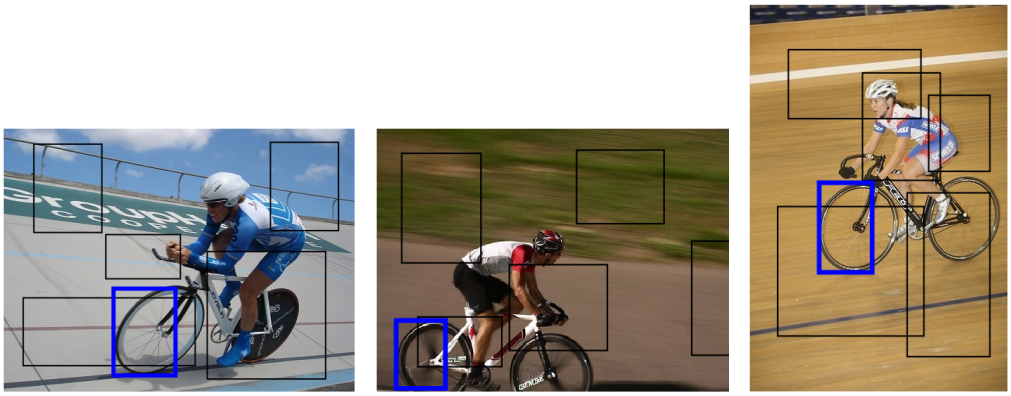

Figure 1: Our system aims at discovering distinctive parts (blue boxes) from a set of regions (black boxes) randomly extracted from images of a category.

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 parts is expected in the images of the category. These definitions are implemented into an objective function which is optimized during a learning stage. The objective function relies on a *match* function which automatically discovers and relates model's parts to image regions. Training can be achieved from a set of images describing the category to be recognized, without having to provide any extra annotations. In particular, bounding boxes revealing objects locations are not necessary. During training, a part classifier is learned in conjunction with the alignment of parts to image regions. In a second time, these classifiers can be used to build a global visual descriptor of images, which combines the signatures of the regions discovered in the image. More precisely, the paper proposes 3 representations: one is obtained by aggregating the Deep CNN signatures of the different image regions, another consists in aggregating the scores of individual part classifiers while the third encodes the distinctive regions of an image with a Fisher vector.

The proposed approach is experimentally validated on three classification datasets. First, Willow [18] aims at classifying 7 human actions in still images, while the goal of Boats Datasets is to classify 5 different categories of

boats. Finally the MIT 67 dataset [19] contains images of 67 types of scenes which are to be recognized. These experiments show that not only the proposed method outperform Deep CNN but also that it offers state-of-the-art results on the very competitive MIT 67 dataset.

The rest of the paper is organized as follows. Related work is presented in Section 2, while Section 3 provides details on the proposed system that learns, aligns, and encodes distinctive parts. Finally, the experimental validation is given in Section 4, before concluding the paper.

2. Related Work

Image classification. has received a large attention from the computer vision community, *e.g.* see the abundant literature related to the Pascal VOC [20] and ImageNet [21] challenges. A large part of the modern approaches follow the bag-of-word model [2], composed of 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 representation, 4) training and classification of pooled image descriptors for the purpose of object recognition. Several studies evaluated the influence of the first step: the low level features *e.g.* gradient, shape, color, and texture descriptors, such as [22], while other proposed combining different levels (low - mid - high) of information [23]. Regarding the second step: image encoding; Fisher vectors [5] were considered as achieving state-of-the-art performance, in many cases. The third, pooling, step is also shown to provide improvements, and spatial and feature space pooling techniques have been widely investigated [24, 4]. Moreover, [3, 14] have recently proposed two different strategies for embedding spatial information into the bag-of-words framework. Finally, regarding the last step of the pipeline, discriminative classifiers such as Support Vector Machines (SVM) are widely accepted as the reference in terms of classification performance.

During the last months, the deep CNN approaches have been successfully applied to large-scale image classification datasets, such as ImageNet [21] [1],

obtaining state-of-the-art results significantly above Fisher vectors or bag-of-words approaches. These networks have a much deeper structure than standard representations, including several convolutional layers followed by fully connected layers, resulting in a very large number of parameters that have to be learned from training data. By learning these networks parameters on large image datasets, a structured representation can be extracted at an intermediate to a high-level, depending on the extracted layers [25, 26]. Deep CNN representation have been recently combined with VLAD descriptors [27] or Fisher vectors [9]

Mid-level features. Several authors have shown the importance of adding intermediate representations [28], also referred as the mid-level features, for leveraging the performance. We observe three mains trends of 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 pixels such as superpixels [29, 30], patches [31] or segments [32]. 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 original method was the Deformable Part Model, proposed by [17]. We can also mention the semantic attributes [33, 34] which have received a lot of interest. Within the learned mid-level features techniques, we observe a large variety in terms of learning data utilized. While some feature are based on extra training data such as labeled fragments [35], sketch tokens [36] or pre-trained object detectors [37], most methods use a standard split of training and testing data to learn the distinctive features, as the *structural element patch model* [38] or the *blocks that shout* [11]. Finally, regarding *unsupervised mid-level features*, the work of [39] aims at detecting distinctive patches in an image dataset without any label information.

Learned mid-level features. Our work aims at learning distinctive parts without extra annotations. Therefore, closely related work includes the Deformable Part Model (DPM) [17]. 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 of parts positions as well as the variation of appearance within a category makes the problem different. Our work also bears similarities with [16], which tries to discover the *fragments* that maximize the mutual information between the category and the presence of the fragment in the image. However, [16] suffers from that (i) contrarily to [17], 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, [10] 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 [11] and [12] which both propose algorithms for learning parts that are good representatives of a given category. In the same way, [40] proposed a part localization model leveraging Deep CNN features computed on bottom-up region proposals, by learning part appearance models and enforcing geometric constraints between parts. Our work follows the same objectives, without the localization constraints imposed by [12, 40] and the large computation requirement and unoptimized encoding of [11]. This work finally shows the importance of mid-level information and justifies its use to improve recognition capabilities.

The automatic discovery of distinctive parts is a very active area explaining that some closely related papers have been published since the submission of this article. The very recent work of Parizi *et al.* [41] shares the same objectives than ours and proposes to learn a part-based model by simultaneously learning an image classifiers and a set of shared parts. Three

different recent papers have explored how Deformable Part Models and Deep CNN can be combined: [42] shows that a DPM can be formulated as a CNN, thus providing a synthesis of the two ideas, [43] also integrates the non maximal suppression phase in the CNN architecture, while [44] proposes a deformation-constrained pooling layer designed to learn shared visual patterns and their deformation properties for multiple object classes.

This article is an extension of [45] providing a richer description of the related works, more details and improvement of the method as well as a much more experimental validation.

3. Proposed method

From a general point of view, the overall approach consists in three steps: (i) a learning step during which some category's distinctive parts are discovered, (ii) a representation step in which a global signature of the image is computed, on the basis of parts presence in the image, (iii) a classification step relying on a linear SVM classifier. The originality of the work is in the discovery of category's distinctive parts and their use in the encoding of images (two first steps), which are the subject of this section, and not in the classification step which is the most classic.

The model we propose for representing image categories consists in a collection of K distinctive parts defined by their visual appearance, without any geometric relationships between them. It is expected that positive images (with respect to a given category) contain regions visually similar to these parts (considered as instances of the parts) while there are fewer of them in negative images. The distance from an image to the class is then defined as a function of the set of distances between image regions and parts (*e.g.* using max pooling).

As aforementioned, the main contribution of this paper lies in the method allowing to automatically discover distinctive parts in the images of a given category and thus to learn the model of this category. These parts are further aligned with images regions, which are utilized to produce images signatures.

Signatures are subsequently used in a standard classification framework.

This section first presents our part-based model and its associated cost function, which is to be optimized during learning. In a second time, 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 images signatures can be computed using the learned model.

3.1. Part model and objective function

First, let us 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 or CNN, representation of the region. More details on the description choices are discussed in section 3.4. 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.

As explained before, our model relies on three assumptions: first, the model is supposed to be composed of a set of K different parts. Second, it is expected that each part of the model is present in each positive image. Third, 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 second constraint by introducing the match function $m(r, p)$ associating model parts and image regions, and by imposing that $\forall \mathcal{I} \in \mathcal{I}^+ r \in \mathcal{I}$ 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 the first 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 third 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 [11], we use the LDA technique of [46], which consists in learning once and for all a universal model of negative patches. We note that our method differs largely from the one proposed in [11]: Our description, initialization, and learning methods are totally different. However, we share the same goal of discovering parts representative of a class, as well as using the LDA technique and using a similar encoding of parts response. In practice, the parameter vector w of a part classifier, corresponding to the part p, is defined simply as:

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

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:

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

Learning this model hence 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 [47] to obtain an approximate solution, as explained in the following section. This algorithm was first introduced to solve simultaneously the correspondence problem as well as the pose estimation of 3D and 2D data. In [47], 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.

3.2. Learning with softassign

Now, our main goal is 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 as a permutation matrix. We use the *deterministic annealing* method of [48] 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 rows and columns of the matrix should sum up to 1 (see [47] for more explanations). Therefore, the computation of the match function can be achieved iteratively using the *softmax* formulation:

$$\forall I \in \mathcal{I}^+,\ \forall r \in I,\ m(r,p) = \frac{\exp(\beta \times w^T(p, m^*) \times x_r)}{\sum_{r' \in I} \exp(\beta \times w^T(p, m^*) \times x_{r'})}. \qquad (4)$$

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.

However, we utilized the following formulation, which improves numerical stability. $\forall I \in \mathcal{I}^+$ and $\forall r \in I$,

$$m^\dagger(r, p) = \exp(\beta((w^T(p, m^*) \times x_r) - \max_{\forall r' \in I} (w^T(p, m^*) \times x_{r'}))). \tag{5}$$

In addition, the match matrix m has also to satisfy the doubly stochastic constraints. This can be achieved by using Sinkhorn (see more details in [47]), 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 $\forall r \in I$, $\sum_{p \in \mathcal{P}} m(r, p) \leq 1$. Gold *et al.* [47] turned the inequality constraint into an equality constraint by adding a slack variable [49]. However, unlike Gold *et al.* [47], our problem is not symmetrical. In order to handle the inequality constraint, we add non-linearities to the process by setting to zero the very low values (*inferior to* 10^{-7} *in practice*), of $m^\dagger$ right after its calculation, see Equation 5. Then, the following process is the normalization, which distributes the weights, except for the null values that remain unchanged. Therefore, the normalized match-matrix satisfies the previous constraints: $\forall I \in \mathcal{I}^+, \forall p \in \mathcal{P}, \sum_{r \in I} m^\dagger(r, p) = 1$ and $\forall I \in \mathcal{I}^+, \forall r \in I, \sum_{p \in \mathcal{P}} m^\dagger(r, p) \leq 1$. For example, if a region obtains a very low score for all parts, a column of the match-matrix $m^\dagger(r, p)$ is set to 0. In other words, image regions not matching any parts, *with very low scores*, will not contribute to any parts and while the parameter β is increased the selection will be more strict and more regions will be discarded. This process further allows to speed up the normalizations in the algorithm.

3.3. Providing intitial correspondences between parts and image regions

The learning process allowing to learn distinctive parts by iteratively refining the match function m, as presented in the previous section, is a process requiring to know m from the previous iteration, and hence raises the question of the initialization of m.

It seems reasonable to think that because the optimization process is not convex, the algorithm will perform better if the initial part to regions correspondences already involve discriminative regions. To select these initial discriminative regions, we first extract the signatures x_r of the regions sampled from positive training images. These signatures are then clustered, using K-means. Then, we use again the LDA acceleration of [46] to learn initial classifiers. 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:

$$\begin{aligned} s_j^+ &= \tfrac{1}{|\mathcal{I}^+|} \textstyle\sum_{r^* \in \mathcal{I}^+} w_j^T x_{r^*} \\ s_j^- &= \tfrac{1}{|\mathcal{I}^-|} \textstyle\sum_{r^* \in \mathcal{I}^-} w_j^T x_{r^*}. \end{aligned} \tag{6}$$

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 discriminative 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 $m_0(r, p)$.

3.4. Computing region and image signatures

3.4.1. Image region signatures

First, we would like to comment on the patch signatures x_r, used in the learning process. We note that these descriptors must be compact, *i.e.* no

more than a few thousand dimensions, to allow the learning to be effective. In fact, we remind that each image is represented by a few thousand of these patches, or regions. Therefore, we first used the simple BOW description using $k = 1000$ clusters, as in [12]. Later, following [25], we extracted the seven-th layer of the CNN representation. This intermediate representation offers much higher results, as we can see in section 4, and allows a better comparison to the current best performing methods. Therefore, we build two systems, or pipelines, the standard one or BOW based on SIFT and the CNN-based pipeline.

3.4.2. Image signatures

Once the model is learned, images signatures can be computed using the distinctive parts of the model. Let us denote as I an image to be encoded. We first extract a set of random 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 pool the per part similarities to produce a global signature of the image.

We propose three different pooling/encoding strategies: the Bag-of-parts inspired from [11] and two novel approach so-called the *Fisher-on-parts* and the *CNN-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 over the image. 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:

$$\frac{\sum_{r \in I} w^T(p_j, m) \times x_r}{|r \in I|} \quad \text{and} \quad \max_{r \in I} w^T(p_j, m) \times x_r. \tag{7}$$

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.

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:

$$r^* = \arg\max_{r \in I} w^T(p, m) \times x_r. \tag{8}$$

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.

Encoding images with CNN-on-parts. In this case, regions are encoded with CNN features and scores are obtained for each region of an image, as for the Bag-of-part signature. For each part or the model, the region giving the highest score (see previous paragraphs) is selected and it's descriptor kept. All the descriptors so selected are further concatenated resulting in a $D^\star \times K \times C$. Where $D^\star = 4096$ is the dimension of the CNN descriptor.

4. Experiments

This section presents an experimental validation of the proposed approach. We start by describing the datasets used in our experiments; then we introduce some baseline algorithms used for comparison purposes and give the details of our implementation; finally the performance obtained with the proposed approach are exposed and compared to baselines and state-of-the art algorithms.

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
      $\hat{m}_0^\dagger(r,p) \leftarrow \frac{\hat{m}_1^\dagger(r,p)}{\sum_{p\in \mathcal{P}}\hat{m}_1^\dagger(r,p)}$
    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.

4.1. Datasets

Three classification datasets are utilized to experimentally validate the proposed approach: The Willow actions dataset [18], the Boats Dataset, and the MIT 67 scenes dataset [19].

The Willow actions dataset [18] is a dataset 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 these bounding boxes, as we want to detect the relevant parts of images automatically without any prior knowledge on the scenes.

The MIT 67 scenes dataset [19] 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 2,877 images divided in 5 categories of boats (*e.g.* boating, fishing, merchant ship, tanker, passenger).

In the following, the performance on the three datasets is measured using the mean Average Precision (mAP).

4.2. Comparisons to baseline approaches

Our approach is compared to different state-of-the-art approach of the literature.

On one hand we report results obtained with Bag-of-words and Fisher vectors computed on the dense root SIFT, on the whole image (see [50] for details). In addition, we used Fisher vectors computed on spatial pyramids, using the two first layers *i.e.* 1×1 and 2×2 segments. A SVM classifier is then trained on the train set and applied to the test images, following the standard procedures for such image classification tasks.

We introduce a second baseline inspired by the work of [13]. A CNN is first trained with CAFFE on ImageNet (for experiments on the Willow action dataset) or ImageNet+Places datasets [26] (for the MIT 67 Scenes), and the penultimate layer of the network is used as an image descriptor. The images of the target dataset (*i.e.* MIT 67 or Willow) are then encoded using this CNN-based descriptor and processed with a standard linear SVM classier framework.

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

4.3. Details of our classification pipeline

As explained in the previous section, the proposed algorithm relies on two steps: a first step in where the parts are learned and a second one in which a global signature of the image is computed, using the selected parts. In the first stage, two different encodings of the regions are evaluated (SIFT based bag-of-words and CNN features). In the second steps 3 different encoding/pooling strategies are considered: (i) Fisher-on-parts consisting in computing SIFT-based Fisher vectors on the selected regions, (ii) Bag-of-parts in which the scores of each part classifier are aggregated to form the image descriptor and finally (iii) CNN-on-parts in which the CNN descriptors of each region are concatenated to form the image descriptor (see section 3.4 for a description of these image descriptors).

Once images descriptors are computed they are processed by a linear SVM such as usually done for these classification tasks. We remind that the originality of the work is in the encoding of the image and not in the classification step which is standard.

In the following paragraphs, we give the details of the implementation used in this experimental validation.

Extraction of image regions. For each image, a set of regions is generated by randomly sampling 2,000 regions per image, over the entire image. We note that for the CNN-based pipeline, only 1,000 regions are extracted to save time. The scale and aspect ratio of these regions are randomly chosen, but regions are constrained to have a size of at least 5% of the image size and aspect ratio should belong to $[0.5; 2]$.

Regions descriptors. As said above, two types of regions descriptors considered. For the BOW-based region descriptors, dense SIFT features are extracted within the regions to be encoded, using VLFEAT [51]. We use the default 4 scales, and sample points every 3 pixels. The SIFT features are further square-rooted to get rootSIFT features and the feature dimension is reduced to 80 using PCA, as suggested by [50]. Then each region is charac-

Table 1: Results for the (SIFT) Bag-of-parts showing the influence of the initialization and the optimization processes, on the Willow dataset.

Method	BOP on salient regions	BOP (random init)	BOP (proposed approach)
mAP	0.467	0.460	0.510

terized using a 1,000-dimensional bag-of-word. These choices are standard for this type of problem [50].

Regarding the CNN descriptors, we use the 7-th layer of the CNN proposed by [52], resulting in a 4,096-dimensional vector. For the experiments on the Willow action dataset, we use the standard CAFFE CNN architecture [52] trained on ImageNet. For those on the MIT 67 Scenes dataset, we use the hybrid architecture [26] trained on ImageNet and on the Places dataset. Note that the same description method is used to compute region descriptors within the CNN-on-parts descriptor (such as defined section 3.4).

Parameters of the learning algorithm. Regarding the learning algorithm, we empirically set the parameters as suggested by [47]: $\beta = 0.41$, $\beta_r = 1.245$, $\beta_f = 1.2$, $I_0 = 4$, $I_1 = 30$ (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 $\epsilon = 0.005$.

4.4. Results

In this section, we first comment on the quantitative results then show some qualitative results, i.e. visualization of learned parts, in Figures 4 and 5. As said above, the performance is measured using the mean Average Precision (mAP).

Initial matchings between model parts and image regions. First, we evaluate the impact of the initialization step in the part-learning process, on the Willow dataset. Results are given Table 1. The objective is to measure the

Table 2: Performance of the CNN-based Bag-of-parts descriptors. Left-hand side: using salient regions, Right-hand side: using the proposed learning approach.

Method	BOP on salient regions	BOP (proposed approach)
Willow	0.656	0.766
MIT 67	0.555	0.788

contribution of the initial set of correspondences between parts and regions, such as described in section 3.3, and to compare it with a simple random initialization of the parts/correspondences. If we randomly initialize the match function we observe a mAP of 46.0% (with the SIFT based Bag-of-parts encoding). Adding the proposed initialization (based on salient regions) improves the mAP by 5% (51.0%).

In addition, to prove the usefulness of the proposed model versus a simple selection of discriminative regions, we evaluated the performance obtained by initializing the match function with salient regions (using the method proposed in section 3.3), without performing any subsequent optimization, *i.e.* without learning m but keeping the correspondences between parts and salient regions as they initially are. If we just use the salient regions, the performance drops to 46.7%; we did the same observation for the CNN-based pipeline, see Table 2 and Figure 3.

The experiments demonstrate that the proposed algorithm improves significantly over a simple selection of discriminative parts, and that a good initialization of our algorithm is better than a random initialization of the part to region correspondences.

Convergence. Theses experiments aim at understanding how the match matrix converges towards a sparse matrix hard-assigning regions to parts. Figure 2 represents this convergence process, by showing the ratio $\frac{\sum m_{i,j}^2}{K \times |\mathcal{I}^+|}$, where K is the number of parts and $|\mathcal{I}^+|$ the number of positive images. The ratio should be of 1 in case of hard assignments. We note that there is a consistent

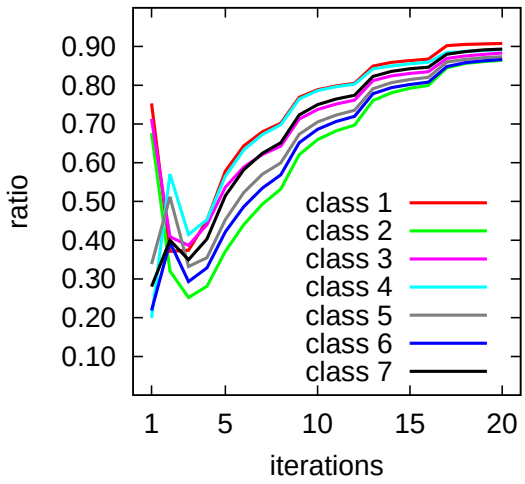

Figure 2: Convergence of the match matrix m for each class of the Willow actions dataset.

drop in the first few iterations, as the initial parts are not (yet) constraint to be generative, *i.e.* a parts should be observed in every positive image of the specific class. Finally, we observe a small step each time the temperature parameter is updated.

Overall, the convergence is behaving as expected.

Bag-of-word based representation. The SIFT-Bag-of-parts and Fisher-on-parts pipelines are then evaluated on the three datasets, see Table 3 and Table 5. 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 note that the (SIFT) Bag-of-parts slightly outperforms the standard Bag-of-word. More interestingly, the proposed Fisher-on-parts representation 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.* [12] which uses the bounding boxes). This explains why we do not provide any comparisons with these methods, as they would be meaningless.

The Boats dataset also shows improvements on both the (SIFT) Bag-of-

Table 3: Results on Willow and Boats dataset. See text for details.

Method	**Willow (mAP)**	**Boats (mAP)**
Bag-of-words [50]	0.500	0.673
Fisher vectors [50]	0.581	0.827
Bag-of-parts	0.510	0.741
Fisher-on-parts	0.614	0.852

parts and the Fisher-on-parts. Specifically, we observe more than 6% and 2% mAP increase over BOW and Fisher vectors respectively.

Concerning MIT 67, we first observe that our (SIFT) Bag-of-parts encoding offers better performances than the Bag-of-word model as well as the Bag-of-parts proposed in [11]. 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 approaches.

Table 4: Results on Willow dataset, using the CNN-based pipeline.

Method	**Willow (mAP)**
CNN on full image	0.763
Bag-of-parts (CNN)	0.766
CNN-on-parts (CNN)	0.816
Bag-of-parts & CNN-on-parts (CNN)	0.819

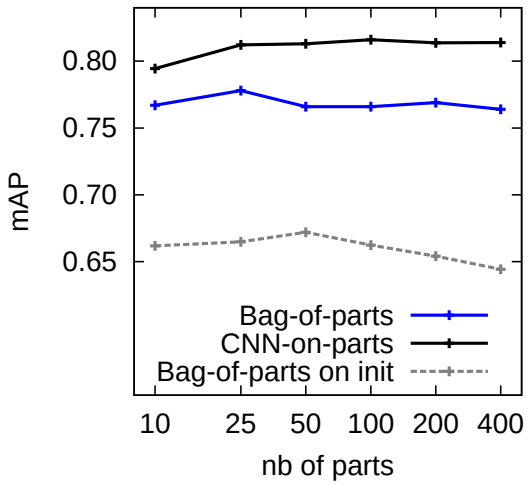

Figure 3: Scores for Willow actions as a function of the number of parts.

CNN-based pipeline. In these experiments, the regions are described by CNN features. First, we evaluate Figure 3 the impact of the number of parts used to describe images, on Willow. It is interesting to note that for (CNN) Bag-of-parts and CNN-on-parts, performances are stable for any number of parts between 25 to 400 parts. Furthermore, utilizing only 10 parts offers reasonable performance. However, if we compute the Bag-of-parts on the initialization parts (salient regions), *i.e.* without learning m, we note that having more than 100 parts slightly reduces the performances.

We also observed a consistent improvement of the (CNN) bag-of-parts and CNN-on-parts over the CNN on the full image, as shown Table 4 and Table 5. For the Willow action dataset the CNN-on-part offers the largest improvement, while the (CNN) Bag-of-part is the best performing method on the MIT 67 scenes dataset. This result supports our observation with the Fisher-on-parts that a foreground pooling effect is very advantageous on Willow actions, while contextual information is better for MIT scenes.

Interestingly, we also observed that combining the two proposed encoding methods – by doing a simple concatenation of the (CNN) Bag-of-part and CNN-on-part representations – makes the performance even better, produc-

Table 5: Results on MIT 67 scenes dataset. See text for details.

Method	**mAP**
bag-of-words [50]	0.345
Fisher vectors [50]	0.550
bag-of-parts of [11]	0.373
our bag-of-parts	0.401
Fisher-on-parts	0.549
Fisher-on-parts based combination	0.600
CNN on full image [26]	0.726
Bag-of-parts (CNN)	0.788
CNN-on-parts (CNN)	0.778
Bag-of-parts & CNN-on-parts (CNN)	0.801

ing performance higher than any reported method to our knowledge (80% of mAP on MIT67).

These experiments show that our descriptors, based on distinctive parts learning, are capable of incorporating mid-level information and produce richer representations.

5. Conclusions

In this paper, we propose a new algorithm to recognize images 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. Based on the part model, signatures are computed to describe images. Finally, the proposed algorithm is validated on three

different datasets on which state-of-the-art performances are obtained.

Acknowledgment

This work is partly funded by the RECONSURVE project.

References

[1] A. Krizhevsky, I. Sutskever, G. E. Hinton, Imagenet classification with deep convolutional neural networks, in: Advances in neural information processing systems, 2012, pp. 1097–1105.

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

[3] J. Krapac, J. Verbeek, F. Jurie, Modeling spatial layout with Fisher vectors for image categorization, in: Proceedings of the International Conference on Computer Vision, 2011.

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

[5] F. Perronnin, J. Sanchez, T. Mensink, Improving the fisher kernel for large-scale image classification, in: European Conference on Computer Vision, 2010.

[6] K. Chatfield, K. Simonyan, A. Vedaldi, A. Zisserman, Return of the devil in the details: Delving deep into convolutional nets, in: Proceedings of the British Machine Vision Conference, 2014.

[7] M. D. Zeiler, R. Fergus, Visualizing and understanding convolutional networks, in: Proceedings of the European Conference on Computer Vision, Springer, 2014, pp. 818–833.

[8] K. He, X. Zhang, S. Ren, J. Sun, Spatial pyramid pooling in deep convolutional networks for visual recognition, in: Proceedings of the European Conference on Computer Vision, Springer, 2014, pp. 346–361.

[9] L. Liu, C. Shen, L. Wang, A. van den Hengel, C. Wang, Encoding high dimensional local features by sparse coding based fisher vectors, in: Advances in Neural Information Processing Systems, 2014, pp. 1143–1151.

[10] S. Maji, G. Shakhnarovich, Part discovery from partial correspondence, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2013, pp. 931–938.

[11] M. Juneja, A. Vedaldi, C. V. Jawahar, A. Zisserman, Blocks that shout: Distinctive parts for scene classification, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2013.

[12] G. Sharma, F. Jurie, C. Schmid, et al., Expanded parts model for human attribute and action recognition in still images, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2013.

[13] M. Oquab, L. Bottou, I. Laptev, J. Sivic, Learning and transferring mid-level image representations using convolutional neural networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014.

[14] J. Krapac, J. Verbeek, F. Jurie, Learning tree-structured descriptor quantizers for image categorization, in: Proceedings of the British Machine Vision Conference, BMVA Press, 2011, pp. 47–1.

[15] H. E. Tasli, R. Sicre, T. Gevers, A. A. Alatan, Geometry-constrained spatial pyramid adaptation for image classification, in: International Conference on Image Processing, 2014.

[16] M. Vidal-Naquet, S. Ullman, Object recognition with informative features and linear classification, in: Proceedings of the International Conference on Computer Vision, 2003, pp. 281–288.

[17] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, D. Ramanan, Object detection with discriminatively trained part-based models, IEEE Transactions on Pattern Analysis and Machine Intelligence 32 (9) (2010) 1627–1645.

[18] V. Delaitre, I. Laptev, J. Sivic, Recognizing human actions in still images: a study of bag-of-features and part-based representations., in: Proceedings of the British Machine Vision Conference, Vol. 2, 2010, p. 7.

[19] A. Quattoni, A. Torralba., Recognizing indoor scenes, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2009.

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

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

[22] N. Pinto, Y. Barhomi, D. Cox, 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.

[23] S. Zheng, Z. Tu, A. Yuille, Detecting object boundaries using low-, mid-, and high-level information, in: Computer Vision and Pattern Recognition, 2007.

[24] Y. Boureau, N. Le Roux, F. Bach, J. Ponce, Y. LeCun, Ask the locals: multi-way local pooling for image recognition, in: Proceedings of the International Conference on Computer Vision, 2011.

[25] M. Oquab, L. Bottou, I. Laptev, J. Sivic, et al., Learning and transferring mid-level image representations using convolutional neural networks, Computer Vision and Pattern Recognition.

[26] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, A. Oliva, Learning Deep Features for Scene Recognition using Places Database., in: Advances in Neural Information Processing Systems, 2014.

[27] Y. Gong, L. Wang, R. Guo, S. Lazebnik, Multi-scale orderless pooling of deep convolutional activation features, in: Proceedings of the European Conference on Computer Vision, 2014.

[28] D. Parikh, Recognizing jumbled images: the role of local and global information in image classification, in: Proceedings of the International Conference on Computer Vision, IEEE, 2011, pp. 519–526.

[29] J. Tighe, S. Lazebnik, Superparsing: scalable nonparametric image parsing with superpixels, in: Proceedings of the European Conference on Computer Vision, Springer, 2010.

[30] R. Sicre, T. E. Tasli, T. Gevers, Superpixel based angular differences as a mid-level image descriptor, in: International Conference on Pattern Recognition, 2013.

[31] B. Fernando, E. Fromont, T. Tuytelaars, Effective use of frequent itemset mining for image classification, in: Proceedings of the European Conference on Computer Vision, Springer, 2012, pp. 214–227.

[32] J. Carreira, R. Caseiro, J. Batista, C. Sminchisescu, Semantic segmentation with second-order pooling, in: Proceedings of the European Conference on Computer Vision, Springer, 2012, pp. 430–443.

[33] Z. Niu, G. Hua, X. Gao, Q. Tian, Context aware topic model for scene recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2012, pp. 2743–2750.

[34] Y. Su, F. Jurie, Improving image classification using semantic attributes, International journal of computer vision 100 (1) (2012) 59–77.

[35] Z. Liao, A. Farhadi, Y. Wang, I. Endres, D. Forsyth, Building a dictionary of image fragments, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2012, pp. 3442–3449.

[36] J. J. Lim, C. L. Zitnick, P. Dollár, Sketch tokens: A learned mid-level representation for contour and object detection, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2013.

[37] I. Endres, K. J. Shih, J. Jiaa, D. Hoiem, Learning collections of part models for object recognition, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2013.

[38] J. Chua, I. Givoni, R. Adams, B. Frey, Learning structural element patch models with hierarchical palettes, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2012, pp. 2416–2423.

[39] S. Singh, A. Gupta, A. A. Efros, Unsupervised discovery of mid-level discriminative patches, in: Proceedings of the European Conference on Computer Vision, Springer, 2012, pp. 73–86.

[40] N. Zhang, J. Donahue, R. Girshick, T. Darrell, Part-based r-cnns for fine-grained category detection, in: Proceedings of the European Conference on Computer Vision, 2014, pp. 834–849.

[41] S. N. Parizi, A. Vedaldi, A. Zisserman, P. Felzenszwalb, Automatic discovery and optimization of parts for image classification, in: International Conference on Learning Representations, 2015.

[42] R. Girshick, F. Iandola, T. Darrell, J. Malik, Deformable part models are convolutional neural networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015.

[43] L. Wan, D. Eigen, R. Fergus, End-to-end integration of a convolutional network, deformable parts model and non-maximum suppression, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015.

[44] W. Ouyang, X. Wang, X. Zeng, S. Qiu, P. Luo, Y. Tian, H. Li, S. Yang, Z. Wang, C.-C. . C. Loy, Deepid-net: Deformable deep convolutional neural networks for object detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 2403–2412.

[45] R. Sicre, F. Jurie, Discovering and aligning discriminative mid-level features for image classification, in: International Conference on Pattern Recognition, IEEE, 2014, pp. 1975–1980.

[46] J. M. B. Hariharan, D. Ramanan, Discriminative decorrelation for clustering and classification, in: Proceedings of the European Conference on Computer Vision, 2012.

[47] S. Gold, A. Rangarajan, C.-P. Lu, S. Pappu, E. Mjolsness, New algorithms for 2d and 3d point matching:: pose estimation and correspondence, Pattern Recognition 31 (8) (1998) 1019–1031.

[48] D. Geiger, F. Girosi, Parallel and deterministic algorithms from mrfs: Surface reconstruction, IEEE Transactions on Pattern Analysis and Machine Intelligence 13 (5).

[49] V. Chvatal, Linear programming. 1983, WH Freeman and Company, New York.

[50] K. Chatfield, V. Lempitsky, A. Vedaldi, A. Zisserman, The devil is in the details: an evaluation of recent feature encoding methods, in: Proceedings of the British Machine Vision Conference, 2011.

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

[52] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, T. Darrell, Caffe: Convolutional architecture for fast feature embedding, in: ACM International Conference on Multimedia, 2014.

Figure 4: Visualization of parts locations for "riding cycle", "playing instruments", and "riding horse".

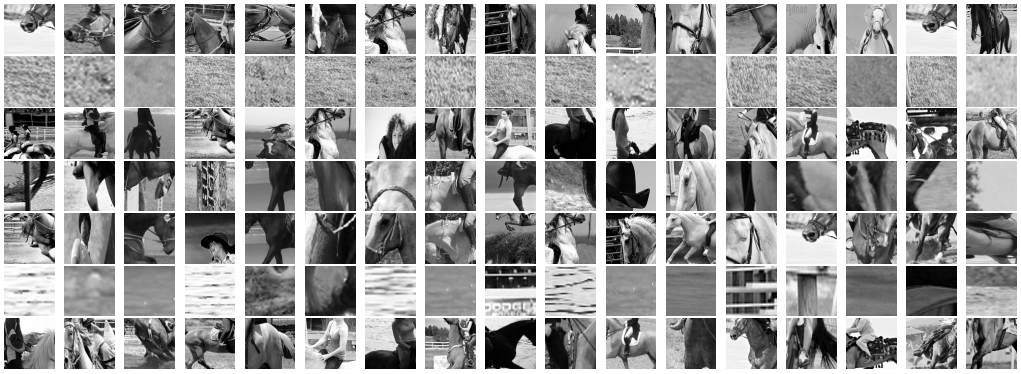

Figure 5: This figure shows the highest scoring regions for a set of parts learned for the *riding horse* action. Each row corresponds to a part.