Context-based pruning for scalable object detection

Marc Masana Castrillo

Abstract

Most existing approaches to object detection scale linearly with the number of classes. We investigate several pruning strategies which result in sub-linear scaling. We exploit fast context classification based on deep features to select a reduced set of detectors. This prevents the unnecessary computation of detectors which are not expected within the context. Results show a linear speed-up in the number of classes on action recognition and a 5 times speed-up for general object detection with a negligible loss of accuracy. In addition, we show how the proposed pruning strategies can be incorporated into the fully connected layers of Deep Convolutional Neural Networks.

Index Terms

Object detection, image classification, deep features, action recognition.

I. Introduction

BETWEEN 10,000 and 30,000 visual object categories exist that we can recognize and detect in images [1]. Leaving aside the complexity of language synonyms and hierarchical relations between words, there is a consensus on which objects we can easily detect in images. We understand scenes and interact with our surroundings by recognizing and locating different objects in the scene. Over the last decade, research has focused largely on recognizing object categories such as cars, persons or bicycles, as opposed to earlier works on more specific characteristics such as car models. The extensive use and development of machine learning techniques has lead to the refinement of object recognition algorithms, making them better at recognizing more complex concepts. These improvements are partially due to the availability of large evaluation datasets from Internet photo-sharing websites [2]–[4]. As the object detection algorithms improve, the scene understanding bottleneck shifts to the scalability in number of object detectors. In this thesis we tackle this issue by proposing methods to speed up the use of specific object detection models on multiple class datasets in order to make more optimal use of computational resources without decreasing performance.

Traditionally, gradient-based object detection has dominated computer vision. Later on, existing edge- and gradient-based descriptors were improved with the Histogram of Gradients (HOG) [5]. This descriptor was then extended for object part detection, leading to the well known Deformable Part-based Models (DPM) [6]. These detection algorithms are capable of representing highly variable object classes. However, Convolutional Neural Networks (CNN) have recently changed Computer Vision significantly, performing extremely well in several classical problems such as object detection, scene classification, face recognition or object pose estimation [7]. These techniques were applied to object detection by means of the Regions with CNN features (R-CNN), introducing object proposals after processing the image through the convolutional layers [8]. This allows extraction of deep features from several sub-images that will go through the fully-connected layers that decide if the requested object is in them or not. Recently, faster and more accurate similar methods such as *fast* and *faster* R-CNN [9], [10] have been proposed, which aim to go towards real-time object detection [9], [10].

There exist several approaches to speeding-up object detectors. Traditionally, most detection algorithms rely on sliding window approaches and try to reduce the number of windows to be executed in order to improve the computational time. In 2008, a branch and bound based approach was proposed for fast object detection over bag-of-word image representations [11]. In 2010, a coarse to fine strategy was proposed to reduce the number of windows by using a multiresolution pyramid [12]. Recently, a fast DPM approach to object detection was proposed based on hierarchical quantitzation and the use of hash tables to generate priority lists [13]. All these methods rely on strategies that try to make object detectors faster. However, very little research has investigated scalability with respect to the number of classes.

In this thesis, we exploit context to obtain a reduced set of detectors that will be executed on each image. For each image, we obtain a context description using deep features. Then, several pruning strategies are used to remove detectors for categories unlikely to be found in the image. This allows for a speed-up per image that scales with the number of classes that we are trying to retrieve. With the increase of the number of objects that must be detected, this method directly tackles the computationally inefficient problem of running a significantly large number of detectors that are doomed to detect nothing under given context conditions.

This thesis is structured as follows. In the next section we give a brief summary of related work and current state-of-the-art, and in Section III we introduce the proposed method's pipeline overview. We describe in Section IV how we can represent

Author: Marc Masana Castrillo, mmasana@cvc.uab.cat
Advisor 1: Andrew Bagdanov, CVC, UAB
Advisor 2: Joost van de Weijer, CVC, UAB
Thesis dissertation submitted: September 2015

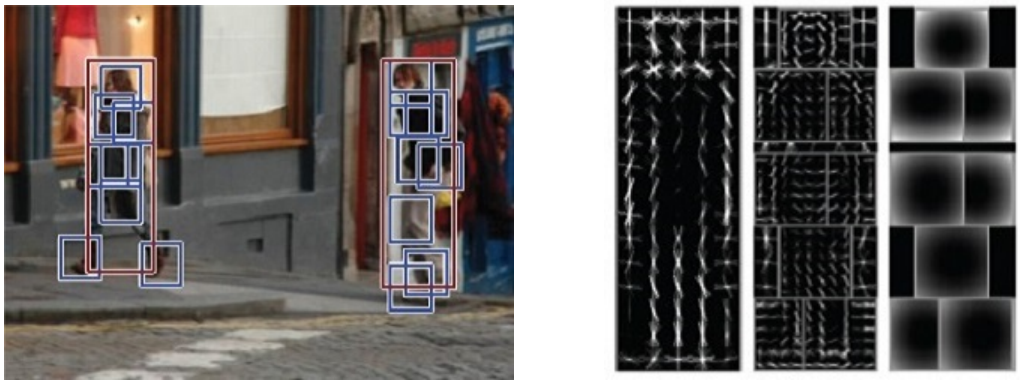


Fig. 1: Detections obtained with a single-component person model (left), and coarse root filter, several higher resolution part filters and spatial model for the location of each part relative to the root (right).[1]

context. In Section V we introduce the context-based pruning strategies. In Section VI we investigate the implications of the proposed pruning strategies for DPM and CNN approaches to object detection. Section VII is dedicated to analyze the experimental results for the proposed pipeline on two well-known datasets. Finally, Section VIII proposes some conclusions and gives some tips on future work.

II. State of the art

The objective of this thesis is to find solutions to the scalability problem of multiple-object recognition.In recent years, the main focus was on recognizing specific objects, usually by generating individual models for each one. However, with the latest improvements in most of those systems, the problem of increasing the number of objects to recognize increases too. In this section we will review the state-of-the-art in visual detection in order to provide a better understanding of our proposals.

A. Object detection

In 2006, the histogram of gradients features were introduced as a new way of improving the already existing edge- and gradient-based descriptors [5]. The idea behind the HOG descriptor is that local statistics about intensity and orientation of gradients can encode the appearance and shape of objects. It is usually implemented as a sliding window approach with overlapping blocks where the HOG feature vectors are extracted. Then, those vectors are combined and used as input for a linear SVM classifier. In the detection phase, a window on all positions and scales is applied across the image and conventional non-maximum suppression is run on the output pyramid to detect object instances. Grids of local normalized HOG features were shown to provide a reduction in false positives on person detection and therefore improve on the performance of existing detectors based on Haar wavelets. A reduction from the original 36-dimensional HOG features to a 13-dimensional feature with almost the same information was proposed in [6] as an improvement for discriminative part-based models.

Performance of object detection methods improved with the introduction of new local and semi-local features, typically robust to occlusions, illumination changes and small deformations. In recent years methods with wavelet-like features [14], [15], locally normalized histograms of gradients [5], [16] or that learn dictionaries of local structures from training images [17] have all been applied to object detection. A significant body of work on deformable models of various types for object recognition exists, including template models [18]–[20] and part-based models [21]–[24]. In 2010, Deformable Part-Based Models (DPM) were introduced for object detection [6]. They use histograms of gradient features (HOG) and introduce a variation to reduce the feature size, as well as using principal component analysis to lower the feature space dimensionality. A sliding window approach is used to obtain a dense set of possible positions and scales in an image, and the latent information in them is used for discriminative training of classifiers. The part-based model score will depend on placing a filter at each of those locations and calculating the deformation cost of the localization of the parts respect to a root filter. The problem of the cost of matching all models of the parts is solved by searching first for the root filter candidate (Figure 1). Then, the scale is changed in order to locate the parts and define a deformation cost function for them. Hard examples are used on the training phase as done in other methods such as Support Vector Machines (SVM) [25] and is quite suitable for training with large data sets where only a fraction of examples can fit in RAM. However, recent increase in computer power has forced us to re-think if we need to find a more optimal way of using DPMs.

Due to their superior representation of highly variable object classes, we use discriminative part-based models in our project. They make use of the location of parts in an object hypothesis to predict bounding boxes for the object. They are also able to detect more than one instance of an object per image. Automatic part labeling has the potential to achieve better performance

[1]Figure extracted from "Object detection with discriminatively trained part-based models" [6].

by automatically finding effective parts. However, the computational time increases with more complex models and scales. We propose strategies to make the application of those models much more efficient.

B. Deep features

In order to detect or identify an object in a still image, the standard approach has been to look for interesting points on the object and extract its features. Many different descriptors have been proposed in the literature. Depending on the interest region detector used some will perform better than others. However, qualities like the discrimination power and the robustness to different types of visual changes or detection errors are always desirable. Among those descriptors, scale-invariant feature transform (SIFT) and speeded up robust features (SURF) are two of the most used ones because of their properties and excellent performance on many datasets [26]. For years these handcrafted features have dominated object recognition. However, recently it was shown that features based on convolutional networks obtain considerably better results.

In 1998, a gradient based learning technique was proposed consisting of a multilayer neural network trained with a back-propagation algorithm [27]. However, computational limitations at the time and the availability of labelled data were not sufficient to make the method suitable for complex computer vision problems. It was not until 2012 that technology caught up and deep features improved state-of-the-art results for object recognition [7]. In the field of object recognition, fine tuning of CNNs trained on large datasets made it possible to improve state-of-the-art performances also for small datasets like PASCAL VOC [28]. Excellent results were achieved on image classification, scene recognition and object detection [28]–[30]. Recently, a very deep convolutional network trained on the ImageNet dataset was introduced [31], [32]. They use more convolutional layers to increase accuracy in large-scale image recognition. The representation depth introduced in this work was shown to be beneficial to the classification accuracy and the state-of-the-art performance on the ImageNet challenge, confirming the importance of depth in visual representations. This are the main reasons why in this thesis we will use the pre-trained CNN from [31] as a starting point for feature extraction.

Most recently, several works have been done relating DPM and CNN. Although DPM are graphical models and CNN are highly non-linear classifiers, the DPM algorithm can be formulated as a CNN by mapping each step of the algorithm to a layer from the CNN [30]. When doing so, it seems only natural to then replace the DPM image features for a learning strategy called DeepPyramid DPM. At the same time, part-based R-CNN [33] and *fast* R-CNN [9] were also proposed as fast frameworks for object detection. In R-CNN, detectors and part models are learnt and enforced by geometric constraints between the parts and with respect to the object frame. *Fast* R-CNN focuses on speeding up the computation time by allowing all network layers to be updated during fine-tuning, as well as using multi-task loss to simplify learning in a single training stage. Both approaches improve the performance on detection accuracy.

C. Action recognition

The task of action recognition consists of pairing each person instance on still images to its corresponding label which describes what that person is doing. Until recently, when CNNs started revolutionizing computer vision, most of the approaches for action recognition were based on bag-of-words frameworks [34]–[36], or used relations or interactions between the person and an object [3], [37]. In [38], the fusion between human poses and CNN feature extraction is used to obtain candidate person proposals. It uses transfer learning methods from the general person DPM to the more specific action model and improves performance for action detection considerably. However, its main disadvantage is the amount of time needed to run each DPM, which creates a scalability problem when the number of classes is large. In this thesis we specifically address the scalability of action detection in large scale data sets.

III. Method overview

The traditional object detection pipeline usually consists of applying as many specific object detectors as classes you are looking for in an image (see Figure 2 top). Increasing the number of classes not only increases the computational time, but also the number of resulting bounding box predictions. When dealing with a large number of specific detectors, the chance of including errors also becomes a problem. Specific detectors can provide wrong predictions if used on images with completely different types of scene. This can result in a significant drop on performance if not addressed properly. Although many works have investigated speeding-up object detectors [11]–[13], little work has addressed the scalability in the number of classes. We propose a computationally inexpensive method based on context that allows pruning of the number of detectors to address the scalability problem in number of classes.

In real images objects are usually correlated to the scene where they are found. Context is useful both for giving hints on the objects that can be found in an image as well as the ones that are probably not there. As an example, when faced with a beach scene you would expect to find people, a boat or a crab. However, there is a lower probability of finding an 18-wheeler truck parked on the sand and you would not expect a wildebeest spending its holidays there. All this information can be exploited in multiple object detection problems.

One way to consider the context of an object is to consider the whole image. Traditionally, the field of image classification has focused on solving this problem. In image classification, the task is to extract the global information of the image and provide

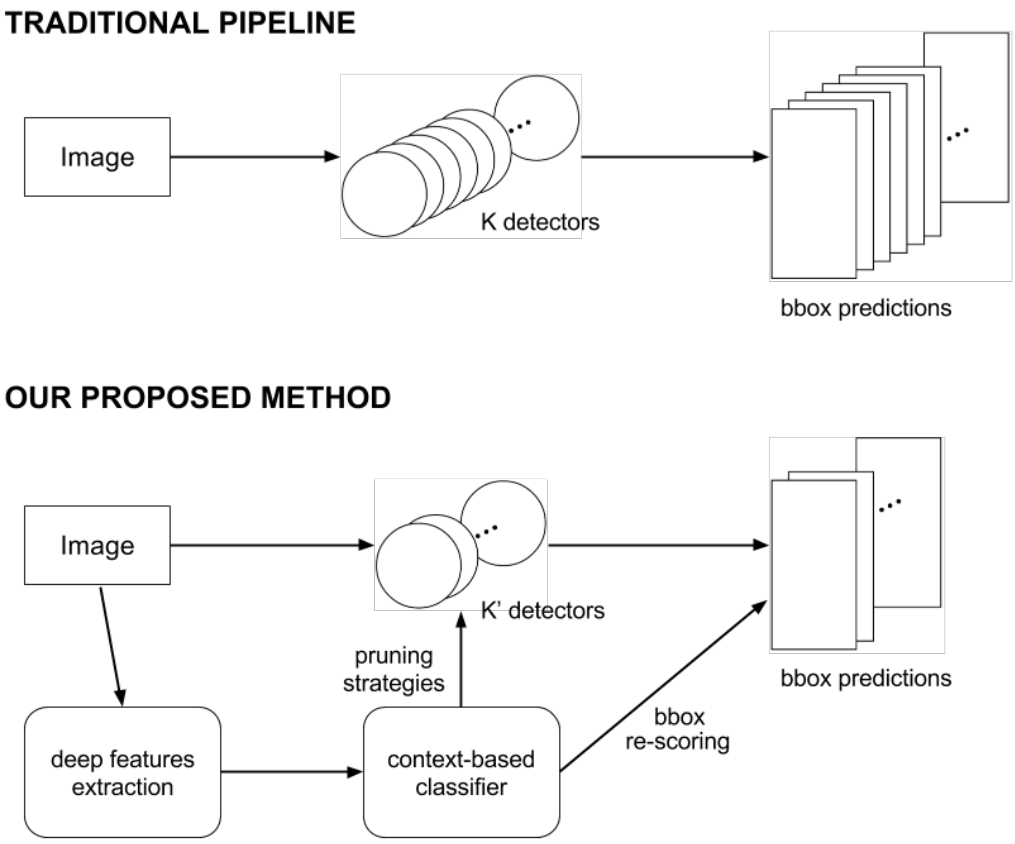


Fig. 2: Traditional pipeline (top) consisting of applying K object detectors, which provide a large number of bounding box predictions. Our approach (bottom) extracts deep features from the image as context information in order to decide which K' detectors will be applied, where $K' < K$. Then the traditional pipeline is followed.

one or more tags (classes) that describe the scene or the events happening in the image. When using context classification, the approach is not supposed to focus on the local relationship between nearby pixels but more on the general global relation of all pixels from the image. Since image classification does not need to deal with object proposals or in-depth correlation of local groups of pixels, it is substantially faster than object recognition. Compared to most object detectors, many inexpensive image classification algorithms exist that provide excellent results [2], [28].

Although we are able to produce better and more precise detectors for different objects, it can be expensive if we want to run them all for every image. In the specific case of Deformable Part-Based Models, each detector is actually quite expensive to run, which means that we can speed up detection on images by carefully choosing which models to apply. Using global features we can have an approximate understanding of which kind of scene we are looking at, and therefore apply only those detectors that are likely to return objects. Global features are used to train a multi-class classifier that predicts which type of scene we are looking at and provides probabilities for each object to be found there. Thus providing an image classification based pruning of improbable detectors. When running an action recognition group of detectors, activities like horse or bicycle riding are not needed to be run on images for which its global features predict as certain scenes, like a living room or a swimming pool.

In this thesis we propose the pipeline depicted in Figure 2 (bottom). We propose a method for multiple-object detection with different context-based pruning strategies for choosing which detectors to run. Note that instead of objects we could be detecting actions or other types of classes depending on the problem to solve. We extract the image context and use an image classification algorithm (see Section IV) to see how probable each object is in the image. These probabilities are then used in the pruning strategies (see Section V) to decide which object detectors will be applied. Context-based pruning takes place before the object detection phase, and is therefore general, in that it is not dependent on the object detector. We will show this by investigating the implications of context on both DPM and CNN pipelines (see Section VI). The main assumption we make is that the computational cost of the context-based pruning part is significantly lower than the cost of object detection. It is desirable that the context-based pruning be computationally faster than running any single object detector. Therefore, if the method manages to prune even one detector, it is computationally worth using it.

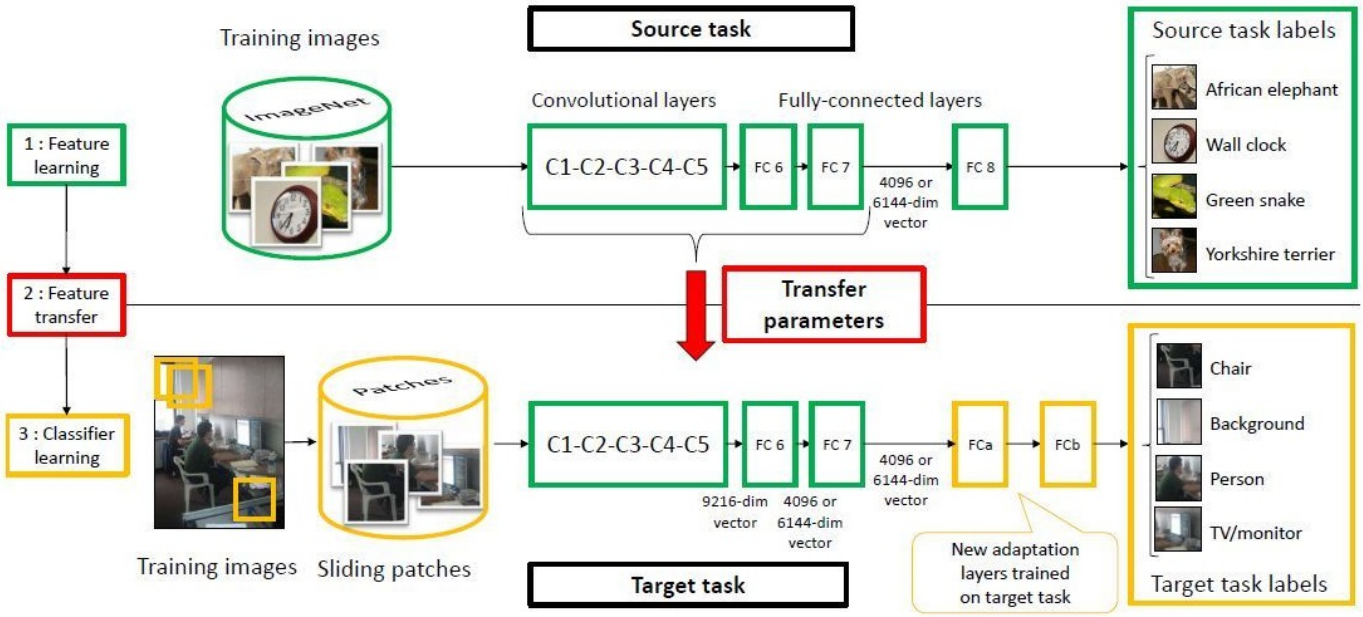


Fig. 3: Transferring parameters of a CNN. The learned parameters from layers C1 to FC7 from the source task (top) are transferred to the target task (bottom). Then, the FC8 layer is substituted by two fully convolutional layers FCa-FCb that act as adaptation layers, which will need to be trained on the target task labelled images.[2]

Let K be the number of classes we want to search for in an image and T_{det} the average time to run a single object detector. Let T_{cbp} be the computational time of the context-based pruning part. The traditional pipeline must run K specific detectors which means that $K \times T_{det}$ time is needed. We assume that context classification and pruning strategy is several times faster than a single object detector, or $T_{cbp} \gg T_{det}$. Therefore, a reduction of the number of detectors results in a similar reduction in overall running time.

IV. Context computation

As discussed above, we use image classification as our context information for pruning detectors. Image classification is the task of predicting the presence (or absence) of a class in an image. Extracting global features from the image provides discriminative information which allows differentiation of those classes that are more likely to appear in that context. Successful classification is critical. Low accuracy, complex fine tuning or slowness of the method can be problematic. However, recent advances in image classification techniques can provide accurate and computationally inexpensive solutions to these problems.

A. Transfer learning and domain adaptation

Recently, deep convolutional neural networks (CNN) have become feasible and have brought a significant performance boost for image classification and object detection. This has been possible thanks to the improvement of specialized high performance computational systems such as GPUs and the newly released large image datasets such as ImageNet [32]. The usage of GPUs not only allows to overcome the computational implementation but also provides an advantage in speed compared to most other pre-existing methods. On the other hand, CNNs demand a large amount of training data. However, transfer learning and domain adaptation towards smaller datasets has been proven to solve this problem with significantly good performance results [28]. The method consists of using a CNN already trained on a large dataset (Figure 3 top). Then, the pre-trained parameters from the internal layers are transferred to a new CNN which will perform the new desired task. Since most CNNs usually have several millions of parameters, those internal layers are used as a generic extractor of mid-level image representation, which allows for the transfer of parameter values from a source task to a target task. Finally, in order to compensate for the differences in image characteristics between the source task and the target task, a couple of fully connected layers are added at the end as domain adaptation layers (Figure 3 bottom). Those new layers have not yet learned the correct parameter values, so they need to be trained on labelled data from the target task.

An example of commonly used pre-trained CNN are the very deep ConvNets proposed by Simonyan and Zisserman [31]. They use 16 and 19 layer models (VGG16 and VGG19) on convolutional networks to increase accuracy in large scale image

[2]Figure extracted from "Learning and Transferring Mid-Level Image Representations using Convolutional Neural Networks" [28].
[3]Figure extracted from https://sites.google.com/site/homepagezhichengyan/home/hdcnn.

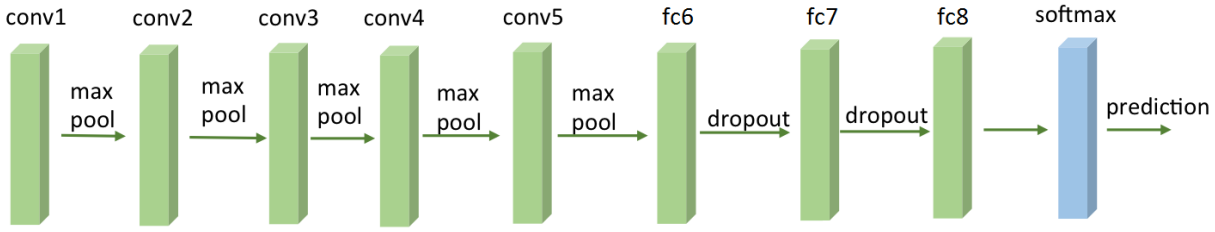


Fig. 4: ImageNet-VGG-16-layer network. Each conv block includes three convolutional layers.[3]

TABLE I: Theoretical comparison between characteristics of classification algorithms for large datasets and high dimensional problems. Discriminant Analysis predictive accuracy depends mainly on the modeling assumptions being satisfied (multivariate normal by class). The SVM prediction speed and memory usage depends mainly on the number of support vectors.

Algorithm	**Predictive Accuracy**	**Fitting Speed**	**Prediction Speed**	**Memory Usage**
Trees	Medium	Fast	Fast	Low
SVM	High	Medium	Variable	Variable
Naive Bayes	Medium	Slow	Slow	Medium
Nearest Neighbor	Low	Fast	Medium	High
Discriminant Analysis	Variable	Fast	Fast	Low

recognition (Figure 4). The CNN architecture is trained on the ImageNet dataset and has a simple pre-processing consisting of subtracting the mean RGB values from each pixel from the image. The image then has to pass 5 stacks of convolutional layers (conv1-conv5 in Figure 4) separated by spatial pooling which is performed by max-pooling layers with stride 2 and a 2×2 window. Each conv1-conv5 stack consists of two to four convolutional layers that use filters with a 3×3 receptive field. This small receptive field over all convolutional layers of the network allows to capture the notion of direction (left/right, up/down and center), as well as reduces the number of parameters. By stacking two layers with 3×3 receptive field without any spatial pooling, we obtain an effective receptive field of 5×5. However, the incorporation of more non-linear rectification layers allows for a more discriminative decision function. And the same happens with three of those layers which can be compared to a 7×7 receptive field. After the stack of convolutional layers, the image goes through three Fully-Connected layers (fc6-fc8 in Figure 4) which have 4096, 4096 and 1000 channels respectively. The last Fully-Connected layer has less channels because it performs the classification for the 1000 classes from the ImageNet dataset. Finally, a last softmax layer is added to the top in order to give the corresponding prediction. It has to be noted that, except the softmax layer, all layers are equipped with the RELU non-linearity rectification function [7]. Since this configuration has proven to provide excellent results on the ImageNet Large-Scale visual Recognition Challange 2014 (ILSVRC-2014), it should be considered as a strong candidate as a pre-trained CNN source task for the transfer learning technique.

B. Adaptive layer substitution

Although the above mentioned transfer learning methods provide good results and are less computationally expensive than training a new and specific CNN, they still have the drawback of requiring training of the adaptation layers. This training supposes a significant computational time investment and also needs further fine tuning of many parameters. Because of this, we propose to use the outputs obtained before the adaptation layers as deep features. The main idea is to substitute those layers for a simpler classification algorithm. This substitution is based on the assumption that the deep features obtained at that CNN level are discriminative enough to be used for image context classification. Because the success of the pipeline is dependent on the quality and speed of the context classifier, we evaluated a variety of image classification algorithms that

TABLE II: Comparison between speed and performance of feature classification algorithms. The results are averaged over 10 executions of each algorithm on 4000 train images and 5532 test images from the Stanford40 dataset (see Section VII). The mean times are given after extraction of the model for the full training set and the prediction time for the full test set. For each image, a 4096-dimensional descriptor has been extracted for classification using the MatConvNet framework and the VGG16.

	Liblinear [39]	**Libsvm [40]**		**K Nearest Neighbour**		**Decision Trees**		
options	linear kernel	linear kernel	RBF kernel	euclidean	minkowski	1 tree	45 trees	280 trees
training time (s)	4.12	107.70	312.42	0.79	0.02	121.80	48.38	305.76
testing time (s)	0.43	191.97	238.46	1465.78	1591.79	0.10	2.71	15.44
test images per sec.	12,865	28	23	3	3	55,320	2,041	358
accuracy (%)	70.0	69.3	63.0	55.6	51.9	36.0	60.1	65.8

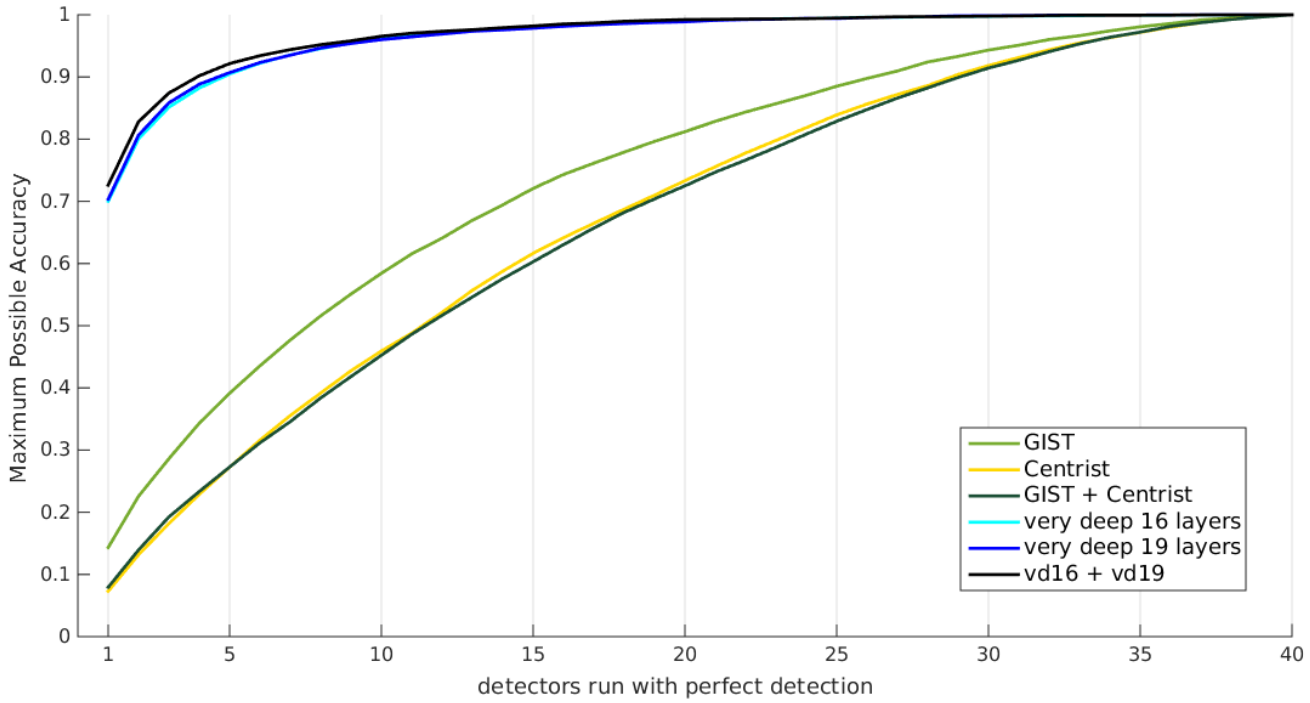


Fig. 5: Context classifier quality for the Stanford40 Action Specific dataset. We consider maximum possible accuracy as having the correct class per each image per number of perfect detectors run. Therefore, when all 40 detectors are used, the accuracy is considered perfect.

should provide a good combination of accurate performance and fast computational time (see Table I). Decision Trees (Tree), Support Vector Machines (SVM) and Discriminant Analysis (DA) seem to theoretically provide good speed while maintaining good performance. However, Nearest Neighbours and Naive Bayes based algorithms are theoretically too slow at detection time for us to consider them further. We also evaluated some of the image classification algorithms in terms of speed and accuracy for the same dataset (see Table II). The study was done with the deep features extracted using the VGG16.

The Liblinear implementation provides accurate computationally inexpensive classification predictions compared to other available methods [39]. Liblinear and Libsvm have very different testing times because Libsvm uses a more complex algorithm that computes the kernel matrix for each train data. Libsvm kernel transformations are not needed for the linear kernel and the Liblinear implementation does not use them, thus making it faster. Therefore, in our pipeline we substitute the adaptation layers for the Liblinear method and no further parameter tuning except for the C-parameter of the SVM is needed for context classification.

C. Context-based classification

The proposed very deep CNN for feature extraction and Liblinear algorithm for classification form the context classification part of our method. The probabilities obtained for each image are used for the object detector pruning. Before explaining the pruning strategies we evaluate if our context-based classification provides results comparable to other known global feature descriptors such as GIST and Centrist [41], [42].

To analyze the potential maximum accuracy reachable by context-based object detector selection we run an experiment with perfect detectors (using the ground truth available for Stanford40). We choose the detectors to run depending on their score when comparing the probability of each action class from the context-based classifier (see Figure 5). For example, in case we consider two detectors per image and the context classification predicts gardening and horse riding classes as the first two classes, only those detectors are considered. The results clearly show that GIST and Centrist are not as suitable for the pruning we want to perform as are the very deep CNNs. Assuming that the detection is a solved problem and we just need to focus on the reduction of detectors to run, deep features provide enough probability of being accurate within the first 5 to 10 ranked classes' scores. Although the best results come from merging the deep features obtained from VGG16 and VGG19, it does not compensate doubling the feature extraction time. Therefore, because of the good results provided and being used in multiple works, we decide to use VGG16 as our proposed feature descriptor.

V. Pruning strategies

Let I be an image which contains $k_1, ..., k_m$ different objects, each belonging to one of K different classes for which we have specific detectors. The traditional pipeline is to apply all of the K specific detectors on the image I in order to obtain a

set of $bb_1, ..., bb_n$ bounding boxes with their corresponding $s_1, ..., s_n$ scores. That means that if each single specific detector has a computational cost c, the cost per image of the traditional pipeline would be:

$$\text{total cost} = c \cdot K \tag{1}$$

A side effect of using this pipeline is that it can produce many bounding boxes from specific detectors whose corresponding class is not present in the image:

$$\text{possible false positives} = \{bb_i \in k_j : class(j) \notin I\}. \tag{2}$$

Therefore, those possible false positives can affect our performance in a negative way sometimes.

Our proposal is based on the assumption that time for image classification is much cheaper than time for detection. Therefore, we spend a computational cost c', where $c \gg c'$, by applying a global classifier to the image I. The output of that classifier will act as an expert that tells us which specific detectors are worth trying. Then, those specific detectors will be applied to the image I as in the traditional pipeline and it will produce another set of $bb'_1, ..., bb'_{n'}$ bounding boxes with their corresponding $s'_1, ..., s'_{n'}$ scores. Then, we know that $n \geq n'$ since we will always run the same or less number of specific detectors. This strategy does not only reduce computational cost, but also prunes some of the resulting bounding boxes from detectors that can produce results in scenes highly unlikely to contain the corresponding object.

Given the probabilities' output by context classification as defined in Section IV, we use the following strategies to decide which detectors are worth trying:

TopN We sort the probabilities for all classes in decreasing order and evaluate the specific detectors for the N most probable ones. Once we have run the specific detectors, we keep the proposed bounding boxes but assign them the global classifier score and evaluate the performance. When keeping the Top1, the speed up for this strategy is N times faster than running all the specific detectors. On the other hand, a TopN strategy means that all specific detectors are run for each image, meaning that the strategy is the same as the existing one and no speed-up is done. Given a dataset with K classes (and thus, K specific detectors) improved with a TopN strategy, the speed-up that we can obtain would be given by the N top classes we want to run instead of the K total ones. This speed-up can be calculated as:

$$\text{speed-up} = \frac{K}{N} \tag{3}$$

and when $N = K$ we have no speed-up. This way, increasing the amount of classes of the dataset can lead to a higher speed-up if the classes can be organized in similarity groups. This is because a large increase of the number of classes maintaining or slightly increasing the number of top specific detectors to run would increase the speed-up ratio, as seen in Equation 3.

ProbsX We obtain the context-based classification probabilities for each class to be found in an image. We define a probability threshold that we consider discriminative enough to decide which specific detectors should be used. All class detectors, whose corresponding probabilities for that image are above the threshold, will be run on the image to retrieve the corresponding bounding boxes. That probability is used as a score in order to make different bounding boxes from different specific detectors comparable. In the case of using DPM for example, each model will have an internal set of values for measuring how good a bounding box is. However, when trying to compare those values, probabilities are much more accurate. With this strategy, the amount of specific detectors used per image is not so clear and it can vary a lot. Therefore, the only way to calculate the speed-up of this strategy is:

$$\text{speed-up} = \frac{K}{\sum_{i=1}^{K} x_i}, \qquad x_i = \begin{cases} 1 & p_i \geq T \\ 0 & p_i < T \end{cases} \tag{4}$$

where K is the number of classes in the dataset, $p_1, ..., p_K$ its corresponding probabilities for that image, and T the threshold. If the chosen threshold is 0, then we fall into the case which applies all specific detectors on the image, and the result is the same as not applying this strategy.

ProbMax This strategy is a slight variation on ProbsX. It exploits the prior knowledge sometimes available of having at least one object in each image. Therefore, even if all the probabilities are below the threshold T on Equation 4, we still run at least one specific detector on the image. The speed up formula for this strategy is a simple modification that forces the strategy to run at least one specific detector, even if it is not very sure of the object being in the image:

$$\text{speed-up} = \frac{K}{\max(1, \sum_{i=1}^{K} x_i)}, \qquad x_i = \begin{cases} 1 & p_i \geq T \\ 0 & p_i < T \end{cases} \tag{5}$$

This modification also solves the problem of sometimes having infinite speed up in some images, which sounds good but actually means not running any detectors in it, therefore not doing a proper detection and reducing our performance.

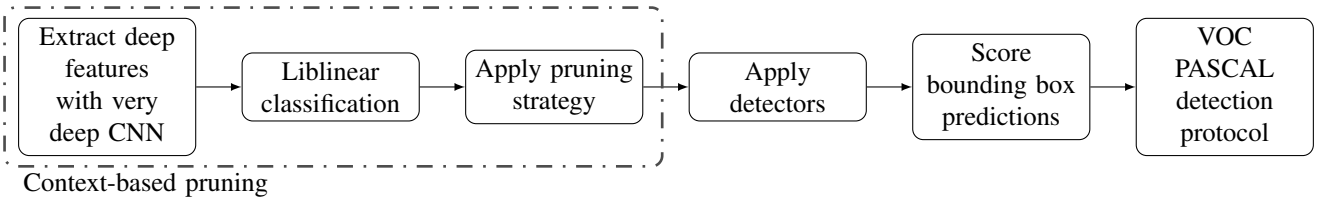


Fig. 6: Object detection pipeline. First we extract deep features which are used in the fast Liblinear classifier to obtain the probabilities of each object being in the scene. Then the pruning strategies to reduce the number of object detectors are applied. After context-based pruning, the traditional pipeline is executed. Then, the corresponding detectors given by the pruning strategy are applied which return the bounding box predictions. Finally, performance is evaluated using the standard PASCAL VOC protocol.

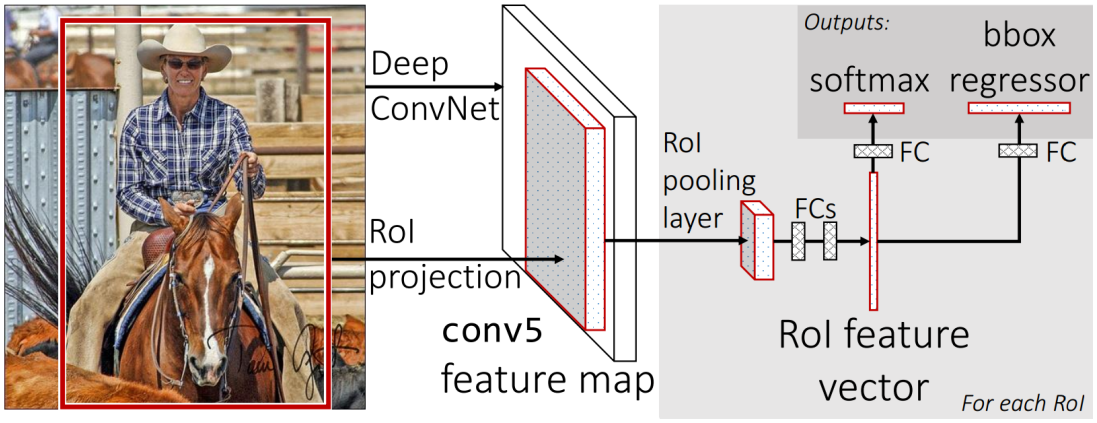


Fig. 7: *fast* R-CNN object detection system overview.[4]

VI. Case studies

The proposed pipeline is general in that it can be used for many of the existing object detection methods. Here we consider two object detection pipelines.

A. DPM: Deformable part-based models

Until recently DPMs were the state-of-the-art object detectors [6]. The DPM automatically learns a structured constellation of parts given a set of training examples of an object. Both the root filter as well as the parts are represented by the HOG descriptor. Application of our proposed pruning strategies to this detector is straightforward and follows the overview in Figure 6. Based on the context scores and the pruning strategy we select a limited set of DPM models which are run for each image.

B. R-CNN: Regions with CNN features

We also investigate the implications of the proposed pruning strategies for CNN approaches to object detection. Incorporating the pruning strategies to deep CNNs is not straightforward since detectors share computation in the networks and are not run separately as in the case of the DPMs. However, it is the last classification layer which scales linearly with the number of classes, and it is exactly there that we propose to incorporate the pruning strategies. It should however be noted that in this case the speed-ups which are obtained are much lower than in the previous case because we now only speed-up the last layer. However, when going to very many class problems this is still expected to be significant [43].

In the last year, the combination of Convolutional Neural Networks with region proposals (R-CNN) [8] or Spatial Pyramid Pooling (SPPnet) [44] has improved object detection performance significantly. R-CNN extracts around 2,000 bottom-up region proposals from the image and computes the features for each region proposal using a CNN. Finally, it classifies those regions using class-specific linear SVMs. On PASCAL VOC 2007, DPM performs at 33.4% while the R-CNN reaches a much better 53.7%. Feature extraction on region proposals is done with the implementation described by Krizhevsky et al. [7], which has two fully connected last layers fc6 and fc7. In R-CNN, they do a layer-by-layer performance analysis and realize that features from fc7 generalize worse than fc6, meaning that 29% of the CNN parameters can be removed without any loss on mAP.

[4]Figure extracted from "Imagenet classification with deep convolutional neural networks" [7].

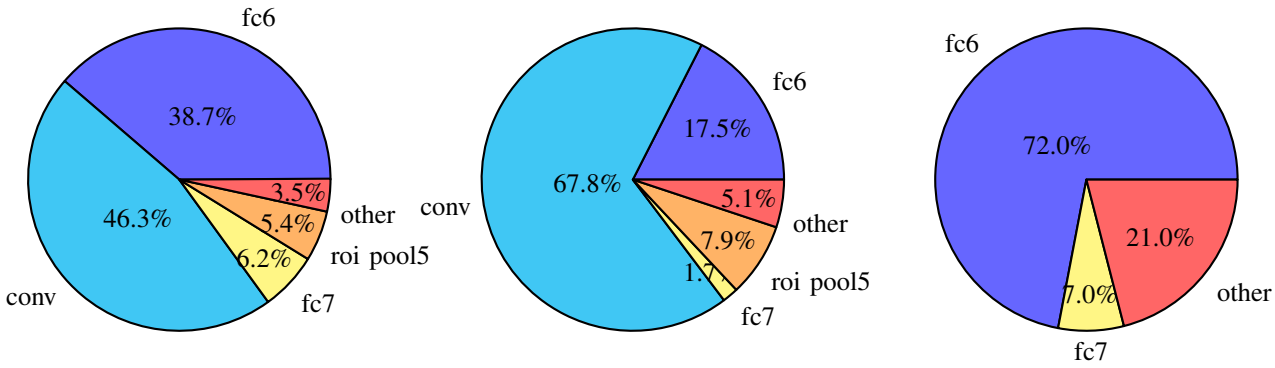


Fig. 8: Timing for VGG16 before and after truncated SVD (left and center respectively). Remaining timing if we only focus on the fully-connected layers (right).

On a CNN that classifies whole-images, the computation time of the fully connected layers is significantly less than the time spent on the convolutional layers. However, when moving onto R-CNN methods, the image is split into several regions of interest given by the region proposals. At this point, the computational time for the fully connected layers increases linearly with the number of region proposals since each of them has to go through the fully connected layers. The *fast* R-CNN method [9] tackles this problem by factorizing the weight matrix W as:

$$W \approx U\Sigma_t V^T \tag{6}$$

using SVD. This allows them to compress the network by replacing the fully connected layer corresponding to W with two fully connected layers corresponding to V and U *without* a non-linearity between them (see Figure 7). This compression is made for a dataset or CNN and provides faster computational times without loss of performance and without needing extra fine-tuning for the two new layers.

In Figure 8, timing for VGG16 before and after applying the above compression is represented. Before SVD, fully-connected layers fc6 and fc7 take 45% of the time. If we focus only on the fully-connected layers, fc6 and fc7 still take most of it with 79% of the time. The remaining 21% is distributed among the rest of the fully-convolutional layers, softmax and bounding box regressor. If we increase the number of classes, the timing spent on the fully-convolutional layer fc8 would also increase. Therefore, the scalability in the number of classes can benefit from a speed-up of the last fully-convolutional layer.

Our proposed pruning strategies can also be used on the weight matrix W from the last fully-connected layer. That last W is a $N \times K$ matrix where N is the number of deep features and K the total amount of classes that can be detected by that CNN. Instead of the CNN specific compression proposed by the *fast* R-CNN, we propose an **image specific** approach. For each image, global deep features are used as described in Section IV in order to obtain probabilities for each class. With these probabilities we will apply one of the proposed pruning strategies on the W matrix and obtain $k_{i_1}, ..., k_{i_m}$ classes that we want to keep and not prune. Then, the new weight matrix W' corresponding to the fully-connected layer will be the reduced version from W without the columns corresponding to the $K - m$ classes to be pruned:

$$W' = W_{1..N, k_{i_1}, ..., k_{i_m}} \tag{7}$$

The here proposed pruning scheme leads to a speed-up of the last layer of the network.

VII. Experiments

In this section we report on a series of action recognition and object detection experiments. We first describe the datasets and experimental protocols used, and then provide qualitative and quantitative results for the context-based pruning of object detectors. For each studied dataset, we will first propose different approaches to each problem and provide baseline results of the different implementations with each pruning strategy. Once we determine the best combination, we will compare our results with state-of-the-art methods.

A. Datasets

Imagenet: 14,197,122 images, 21841 synsets indexed. Imagenet has more than 14 million images with more than 21,000 synsets. A synset is a group of things of the same kind that belong together, and in this case, it refers to words that have similar meanings or subcategories of those words that specialise the meaning. The dataset is organized using the nouns from the WordNet hierarchy (Figure 9), with each node having a mean of 500 images. When using this dataset on Convolutional

[5]Figure extracted from "Learning a tree of metrics with disjoint visual features" [45].

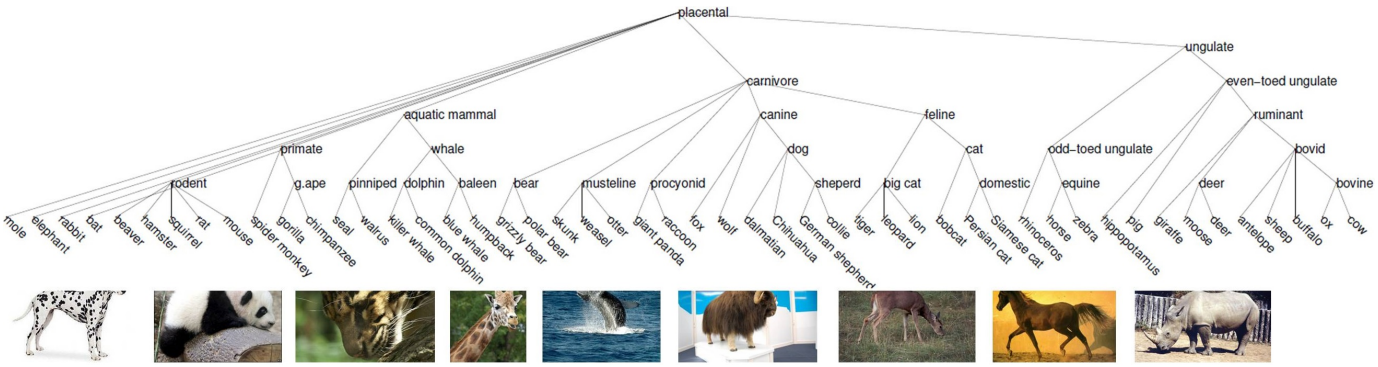


Fig. 9: Examples of class hierarchy from Imagenet. There are around 30,000 images of animals with attributes divided into 50 classes.[5]

(a) Applauding (b) Playing guitar (c) Throwing a Frisbee (d) Drinking (e) Writing on a book

Fig. 10: Examples of human actions in still images from the Stanford40 dataset.

Neural Networks (CNN), discriminative deep features can be learnt in order to explain an image in the same way as you would do with a global descriptor. We use a CNN that has been pre-traind on the 1,000-synset ILSVRC subset which provides the deep features we use for context classification.

Stanford 40 Actions: A dataset for understanding human actions in still images. Our first experiments were done on action recognition on the Stanford40 dataset. This dataset contains 9,532 images divided into 40 categories that depict different human actions such as applauding, playing a guitar, throwing a frisbee, drinking or writing on a book (Figure 10). The dataset is divided into a train set of 4,000 images with 100 samples per class, and a test set of 5,532 images with a variable number of images per class. This dataset contains some priors we can exploit in order to make a more optimal use of the specific detectors as well as a faster implementation. Each image contains only one action by one person. There can be more than one person in the image but only one will be clearly doing the specified action. Therefore, the recognition task is much simpler since we only have to guess one action. In case we have two very probable actions taking place in the image, we know we can just discard the second best guess. Also, if there is no clear action taking place, at least we can predict the less worse probability.

The PASCAL Visual Object Classes Challenge 2007. The VOC 2007 dataset is split into a 2,501 image training set, a 2,510 image validation set and a 4,952 image test set. Once the training parameters have been learned, the training and validation set are usually merged into a trainval set for training the data one last time before testing. This dataset is composed of still images which may contain one or more instances of 20 classes such as car, dog, person, plane or boat (see Figure 11).

(a) Cow (b) Boat (c) Sofa (d) Car (e) Bird

Fig. 11: Examples of still images from the Pascal VOC 2007 dataset.

There is an approximately equal distributions of images and objects by class across the training/validation and test sets. In total there are 9,963 images, containing 24,640 annotated objects. There are 6,301 objects on the training set, 6,307 objects on the validation set and 12,032 objects on the test set. This means that the training and validation set has a mean of 2.52 objects per image, while the test set has a mean of 2.43 objects per image. This dataset, as opposed to Stanford40, requires predictions based more on the per-object-probability rather than focusing on the best prediction per image. The fact that we are sure that an object most probably appears in an image does not mean we can discard other object classes.

B. Evaluation protocol

The precision-recall (PR) curve is the relationship between how relevant the retrieved results are (precision) and how many relevant results are returned (recall). We define those performance metrics as:

$$\text{precision} = \frac{TP}{TP + FP} \tag{8}$$

$$\text{recall} = \frac{TP}{TP + FN} \tag{9}$$

where TP is the number of relevant instances retrieved, FP is the number of irrelevant instances retrieved and FN is the number of instances missed. The PR curve is computed with the above metrics on the retrieved images of the system you want to evaluate, and you can put a threshold on how many images you want to evaluate. If the threshold is high, it will retrieve few examples and the system will be more strict. By contrast, if the threshold is relaxed, the system would return more instances. Note that because of that the precision may not decrease with recall. Therefore, analyzing how precision and recall change depending on the threshold is a good way to evaluate the performance of a system. In order to compare PR curves a single number (the area under the PR curve) provides a more quantitative way of comparison than plotting the curves. The common metric for that is the average precision (AP), which is defined by area under the PR curve:

$$\int_0^1 p(r)\,\mathrm{d}r \quad \Longrightarrow \quad \sum_{k=1}^{N} p(k)\,\triangle r(k) \tag{10}$$

where p are the precision values and r the recall values of the PR curve. However, the standard PASCAL detection protocol uses a modified version of the AP metric by doing an interpolation of the PR curve:

$$\text{average precision} = \sum_{k=1}^{N} \max_{\tilde{k} \geq k} p(\tilde{k})\,\triangle r(k) \tag{11}$$

The intention in interpolating the PR curve in this way is to reduce the impact of the "wiggles" in the PR curve, caused by small variations in the ranking of examples [2]. Since almost all work is compared with this metric, it makes sense that we follow the standard PASCAL detection protocol in order to evaluate the performance of the action recognition and object detection. This evaluation protocol also computes the mean average precision (mAP) of all classes in order to provide a single performance value for the whole dataset across all classes.

C. Action Recognition

For the action recognition experiments we choose the Stanford 40 Actions dataset to evaluate our proposed context-based method. We propose to apply our context-based pruning method on different existing state-of-the-art DPM pipelines.

Baseline performance analysis. We first try our proposed method on a direct-specific approach, which consists of learning a detector for each action class. We use the already trained DPMs from Kahn et al. [38], with a 37.6% mAP. This DPMs have been already trained on the 40 action classes from the Stanford40 dataset. We can use the bounding boxes obtained when using this DPMs but the scores are not directly usable. Therefore, we keep the bounding boxes from the DPMs and add the context-based classification probabilities as the corresponding scores. This pipeline setup provides a slightly better 38.2% mAP, which we will use as our baseline for comparing with the proposed pruning strategies.

In Table III we show the result of applying our proposed pruning strategies to the direct-specific DPM approach.

The direct-specific approach runs all 40 action specific detectors on each image and chooses the best bounding box scores as prediction. However, as the number of classes increase, the time needed to run every DPM for each class increases linearly too. Therefore, we also analyze performance using a single general-person DPM for the detection and the already mentioned context-based probabilities for classification of the bounding box into the corresponding action. As baseline results we will compare with the VOC 2007 person grammar DPM, the VOC 2010 person grammar DPM and Kahn et al. person DPM. Furthermore, we propose the other following modifications on the proposed pipeline:

- context: first use the general single person detector to retrieve the most suitable bounding box to then apply the context-based classification explained in Section IV on the bounding box. This only uses one DPM and the context-based classifier from our proposed method.

TABLE III: Baseline for Stanford40 DPM. Experiments with 40 action specific DPM detectors using context-based classification probabilities as bounding box scores.

pruning strategy	Baseline	TopN	ProbsX	ProbMax
best mAP	38.2	38.8 (+1.2%)	38.2 (±00.0%)	38.8 (+1.2%)
speed	1x	40x	4x	40x
mAP (10x speed)	-	38.0 (-0.5%)	37.9 (-0.8%)	37.9 (-0.8%)
mAP (20x speed)	-	38.1 (-0.3%)	36.9 (-3.4%)	37.3 (-2.4%)
mAP (40x speed)	-	38.8 (+1.2%)	34.0 (-11.0%)	38.8 (+1.2%)

TABLE IV: Experiments for Stanford40 DPM. The baseline uses the context-based probabilities as score for the bounding boxes of the 40 specific action detectors applied. All other results are 40x faster by only using a single DPM per image.

		single person grammar DPM			our proposals			
	Baseline	VOC 2007	VOC 2010	Kahn et al.	pruning strategies	context	bbox	context+bbox
mAP	38.2	26.8	28.6	32.2	38.8	31.3	37.0	42.9

- bbox: first apply the context-based classification to decide which is the most probable action to be taking place in the image. Then, apply the corresponding DPM model for that action to obtain the bounding box. Finally, extract the deep features from the bounding box and apply the context classifier to obtain the score. This strategy also uses one DPM and the almost computationally inexpensive context-based classifier.
- context+bbox: same as previous one but merging the score from the bounding box with the score from the context-based classifier from the whole image. This strategy uses one DPM and two times the context-based classifier. However, the computational time is still less than using two DPMs.

All proposed modifications of the pipeline provide at least similar results as the best single person grammar DPM from the ones proposed (see Table IV). Using the context-based features extracted from the bounding box performs at a similar level to using a single person detector. However, extracting the whole image context and the specific bounding box context, and merging their scores outperform all proposed methods.

Comparison with the state-of-the-art. After evaluating our proposed methods, we choose the best one and compare it with the existing state-of-the-art. Although our method is for pruning of detectors and not for classification, the context-based classification part has been shown to provide good results. We propose to use the same pipeline but, instead of applying the non-pruned detectors, we assign the class with the higher probability as the prediction for that image's class. The probabilities are directly taken from the context-based classifier. Results show that our results are better than state-of-the-art for the Stanford40 dataset until recently, when the Transfer Learning approach in [38] was proposed (see Table V). Regarding detection comparison with the state-of-the-art we face a similar situation as for classification. However, in this case, our method's approach performs at a similar level to the state-of-the-art (see Table VI). Our method is only outperformed by the Transfer Learning approach. However, we expect that our pruning strategies can also be applied to speed up the Transfer Learning pipeline.

D. Object Detection

For the object detection experiments, we evaluate our proposed context-based pruning method on the PASCAL VOC 2007. We use the pipeline explained on Figure 6 with the DPM pipeline and *fast* R-CNN pipeline.

DPM pipeline. We will use as a traditional pipeline the DPMs from the PASCAL VOC Challange by Felzenszwalb et al. [6][6]. They are pre-trained DPM models for the 20 classes of the PASCAL VOC dataset. These models are trained using a discriminative method that only requires bounding boxes for the objects in an image. The approach leads to efficient object detectors. Therefore, our first experiment was setting up the traditional pipeline (see Figure 6, outside the dotted box). This

TABLE V: State of the art for classification on Stanford40 dataset

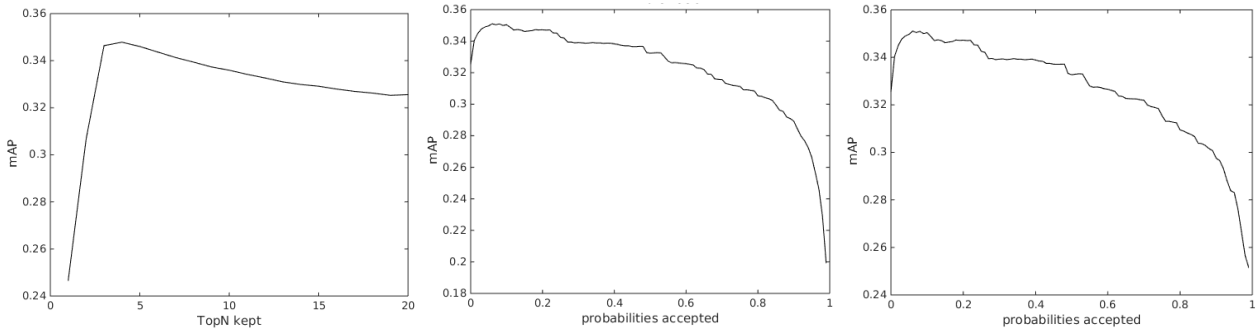
	Object Bank	LLC	Sparse Bases	EPM	CF	SMP	Places CNN	Imagenet	Hybrid CNN	TL	Our method
mAP	32.5	35.2	45.7	42.2	51.9	53.0	42.9	54.9	55.3	75.4	65.0
year	2010		2011	2013			2014				2015

TABLE VI: State of the art for detection on Stanford40 dataset.

	CN-HOG [36]	General-person	Direct-specific	TL	Our method
mAP	27.5	39.7	37.6	45.4	42.9
year	2013	2014			2015

TABLE VII: Baseline for PASCAL VOC 2007 DPM. Bounding box with their respective score and with the context-based classification score results, and same experiments with our proposed pruning strategies.

pruning strategy	Baseline		TopN		ProbsX		ProbMax	
scoring method	bbox	context	bbox	context	bbox	context	bbox	context
best mAP	32.5	24.8	34.8 (+7.1%)	24.8 (±00.0%)	35.1 (+8.0%)	24.8 (±00.0%)	35.1 (+8.0%)	24.8 (±00.0%)
speed	1x		5x	5x	6.5x	5x	5.5x	4x
mAP (10x speed)	-		30.7 (-5.5%)	21.5 (-13.3%)	32.5 (±0.0%)	22.6 (-8.9%)	32.4 (-0.3%)	22.5 (-9.3%)
mAP (20x speed)	-		24.7 (-24.0%)	17.9 (-27.8%)	24.6 (-24.3%)	17.2 (-30.6%)	25.2 (-22.5%)	18.1[7](-27.0%)



(a) Top N (b) ProbsX (c) ProbMax

Fig. 12: Performance of our proposed pruning strategies depending on the PASCAL VOC DPM pipeline.

first experiment will be the baseline of our methods. It yields 32% mAP by applying all 20 DPMs on each image. We will also set that computational time as our reference for further comparisons.

For the rest of the experiments, before the traditional pipeline we compute our context-based pruning part (see Figure 6, dotted box). We first extract the deep features of each image using the VGG16 network. Then, we use that context information as an input for an SVM classifier with linear kernel (Liblinear). Since there is more than one object per image, instead of a multi-class classifier, a One vs All approach is used. 20 classifiers are learnt from the train set. The C parameter is then tuned using the validation set. Then the training of the SVM is done again using the learned C-values on the train and validation sets. Once the final classifiers have been learned they are applied on the test set. In order to all of them to be comparable we need to retrieve probabilities. Once the probabilities for each class to be in the image are computed, the pruning strategies proposed on Section V are applied.

In Table VII we present the best results corresponding to applying the pruning strategies to the DPM pipeline. All pruning strategies improve the existing results for the DPM pipeline with at least a 5x speed-up. Although the ProbsX and ProbMax strategies achieve similar best performance, the ProbsX strategy is a bit faster. Furthermore, the ProbsX strategy achieves the same performance as the DPM standalone pipeline, but 10x faster. These experiments also show that when pushing the system to its limit (since this dataset has 20 classes, that means a 20x speed-up), the performance drops by ¼.

Since re-using the context-based classification probabilities for the bounding box prediction scores provides better results on action recognition, we also try it on object detection. As shown on Table VII, switching the scores from the bounding boxes obtained by the DPMs to the probabilities from the context-based classification does not improve results. The performance drops by a 23.7% but the pruning strategies are still able to work as expected. Although the results are lower than the state-of-the-art, the pruning strategies provide a 5x speed-up, and around a 10% mAP performance drop for a 10x speed-up.

To better analyze on how the method performs depending on the threshold chosen, we plot mAP as a function of N and probability threshold (see Figure 12). The TopN strategy (Figure 12a) starts with a very low performance when only one classifier per class is run. That is expected in this dataset since there is a mean of 2.5 objects per image, and in most cases those objects are of different class. For the same reason, running the Top3 to Top5 context classification ranked detectors performs almost as the baseline, but with a 6.6x to 4x speed-up, respectively. From then on, running more detectors appears to increase the number of false positives. This leads to a decrease in performance until all detectors are used and we hit the baseline. In conclusion, for the TopN strategy on this dataset, speeding up too much decreases the number of retrieved objects by omision, but a 10x speed-up improves performance by decreasing the number of errors. In both ProbsX and ProbMax strategies a similar behaviour is evident (see Figures 12b and 12c). In this case, the lower probabilities we accept, the more detectors are run, meaning that the baseline is comparable to accepting all probabilities higher than 0.00. As we reduce the number of accepted probabilities, performance increases because of the reduction in the number of detectors used. In both cases

[6]Release version 5 from http://people.cs.uchicago.edu/~rbg/latent-release5/ [46].

[7]ProbMax with context scores can only speed up to 18.9x.

TABLE VIII: Baseline results for PASCAL VOC R-CNN. Speed up is only for the fc8 layer.

pruning strategy	Baseline	Top N	ProbsX	ProbMax
mAP (1x speed)	60.0	60.0 (±00.0%)	60.0 (±00.0%)	60.0 (±00.0%)
mAP (5x speed)	-	58.5 (-2.5%)	59.5 (-0.8%)	59.5 (-0.8%)
mAP (10x speed)	-	49.2 (+18.0%)	50.0 (-16.7%)	49.9 (-16.8%)
mAP (20x speed)	-	35.6 (-40.6%)	34.3 (-42.8%)	36.0 (-40.0%)

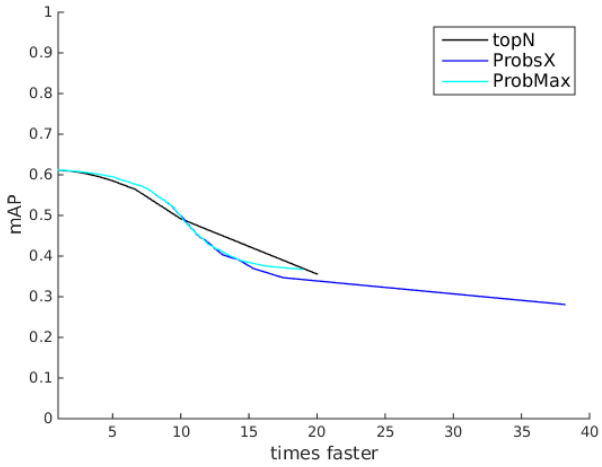


Fig. 13: R-CNN speed up for DPM object proposals using pruning strategies.

the best performance reaches 35.1% mAP by accepting probabilities higher than 0.06. After that, accepting fewer probabilities slowly leads towards too few detectors being used and a decrease in performance similar to the one in TopN.

R-CNN pipeline. The *fast* R-CNN pipeline needs object proposals for each image, which we compute using the DPMs and also the state-of-the-art Selective Search implementation [47]. In Table VIII we present the results of applying both object proposal methods with the *fast* R-CNN pipeline. For the DPMs, we observe that using pruning strategies gives a 5x speed-up with very small loss in performance. However, unlike in the DPM pipeline experiments, when the system is brought to a faster speed, the mAP starts dropping a bit for a 10x speed-up, and dramatically for a 20x speed-up.

In Figure 13 we plot mAP as a function of the thresholds of the pruning strategies. For DPM object proposals, the TopN strategy has a more linear behaviour than the ProbsX and ProbMax strategies. All strategies do well with small speed-ups of between 1x and 7x, while keeping the baseline performance without much loss. However, larger speed-ups start decreasing performance more until about 15x. At this point, it seems that the system stabilizes enough to be able to maintain a similar performance from 15x to 20x.

As seen in Section VI, the speed-up with our proposed context-based pruning method is only located on the fc8 layer of the pipeline. Since our strategy only speeds up the fc8 layer, it does not result in significant overall speed-up for only 20 classes. However, it is clear that if a similar pipeline speed-up can be achieved for a much larger number of classes, the time saved on the fc8 layer would be significantly much larger too. Therefore, the proposed context-based pruning is worth exploring for problems with a larger number of classes in order to solve the scalability problem on the number of classes for time computation.

VIII. Conclusions

In this thesis we investigated the problem of scalability in the number of classes. Most state-of-the-art approaches to action recognition and object detection languish from that problem. In such pipelines, lots of detectors are run in each image, which waste computational time. Instead, we proposed extracting context from images and use it to prune detectors that are unlikely to be found in the image. We evaluated fast methods for context classification and proposed three context-based pruning strategies. Of the three evaluated strategies we found ProbMax to best balance speed-up and accuracy. We show that the scalability problem from object recognition can be alleviated, opening doors to faster and larger-scale object detection.

We evaluated our method on the Stanford40 action recognition dataset and the PASCAL VOC 2007 object detection dataset. For action recognition, our proposed pruning strategies slightly improved performance while providing a 40x speed-up. Also, the proposed modification to use a single DPM and merge global and bounding box specific scores for prediction ranking competes with other recent state-of-the-art methods. We showed that the combination of DPMs for object detection with the

more global deep feature scene recognition provides a considerable speed-up directly proportional to the number of classes to recognize.

Despite obtaining a speed-up of 5x at the cost of less than 2.5% loss in performance, the pruning strategies on the DPM object proposals with the *fast* R-CNN pipeline only reduce computational time for fc8 layer, which is only of little impact on the overall computational time of the object detection pipeline. However, the importance of fc8 becomes more relevant for very large number of classes datasets.

For future work, we will perform further experiments with larger datasets with hundreds or thousands of classes. We expect to refine our method so that it can be used for a much larger number of classes. Also, we will apply the pruning strategies with other object proposal algorithms or to other traditional pipelines which can benefit of the speed-up, as well as the small boost in performance.

ACKNOWLEDGMENT

I acknowledge my advisors Andrew Bagdanov and Joost van de Weijer for all their ideas and support and their contribution in this project, and Fahad Shahbaz Khan for kindly providing the DPM models. I would also like to thank my family for their unconditional support, and Ricard, Corina, Alba, Cristina and Marta for always being there for me. The GPUs used in this research were generously donated by the NVIDIA Corporation.

REFERENCES

[1] I. Biederman, "Recognition-by-components: a theory of human image understanding." *Psychological review*, vol. 94, no. 2, p. 115, 1987.
[2] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International journal of computer vision*, vol. 88, no. 2, pp. 303–338, 2010.
[3] B. Yao, X. Jiang, A. Khosla, A. L. Lin, L. Guibas, and L. Fei-Fei, "Human action recognition by learning bases of action attributes and parts," in *Computer Vision (ICCV), 2011 IEEE International Conference on*. IEEE, 2011, pp. 1331–1338.
[4] G. Griffin, A. Holub, and P. Perona, "Caltech-256 object category dataset," 2007.
[5] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 1. IEEE, 2005, pp. 886–893.
[6] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 32, no. 9, pp. 1627–1645, 2010.
[7] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.
[8] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*. IEEE, 2014, pp. 580–587.
[9] R. Girshick, "Fast r-cnn," *arXiv preprint arXiv:1504.08083*, 2015.
[10] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *arXiv preprint arXiv:1506.01497*, 2015.
[11] C. H. Lampert, M. B. Blaschko, and T. Hofmann, "Beyond sliding windows: Object localization by efficient subwindow search," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*. IEEE, 2008, pp. 1–8.
[12] M. Pedersoli, J. Gonzàlez, A. D. Bagdanov, and J. J. Villanueva, "Recursive coarse-to-fine localization for fast object detection," in *Computer Vision–ECCV 2010*. Springer, 2010, pp. 280–293.
[13] M. A. Sadeghi and D. Forsyth, "30hz object detection with dpm v5," in *Computer Vision–ECCV 2014*. Springer, 2014, pp. 65–79.
[14] C. P. Papageorgiou, M. Oren, and T. Poggio, "A general framework for object detection," in *Computer vision, 1998. sixth international conference on*. IEEE, 1998, pp. 555–562.
[15] P. Viola and M. J. Jones, "Robust real-time face detection," *International journal of computer vision*, vol. 57, no. 2, pp. 137–154, 2004.
[16] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.
[17] E. J. Bernstein and Y. Amit, "Part-based statistical models for object classification and detection," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 2. IEEE, 2005, pp. 734–740.
[18] T. F. Cootes, G. J. Edwards, and C. J. Taylor, "Active appearance models," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 23, no. 6, pp. 681–685, 2001.
[19] J. Coughlan, A. Yuille, C. English, and D. Snow, "Efficient deformable template detection and localization without user initialization," *Computer Vision and Image Understanding*, vol. 78, no. 3, pp. 303–319, 2000.
[20] A. L. Yuille, P. W. Hallinan, and D. S. Cohen, "Feature extraction from faces using deformable templates," *International journal of computer vision*, vol. 8, no. 2, pp. 99–111, 1992.
[21] Y. Amit and A. Trouvé, "Pop: Patchwork of parts models for object recognition," *International Journal of Computer Vision*, vol. 75, no. 2, pp. 267–282, 2007.
[22] M. C. Burl, M. Weber, and P. Perona, "A probabilistic approach to object recognition using local photometry and global geometry," in *Computer VisionECCV98*. Springer, 1998, pp. 628–641.
[23] D. Crandall, P. Felzenszwalb, and D. Huttenlocher, "Spatial priors for part-based recognition using statistical models," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 1. IEEE, 2005, pp. 10–17.
[24] R. Fergus, P. Perona, and A. Zisserman, "Object class recognition by unsupervised scale-invariant learning," in *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, vol. 2. IEEE, 2003, pp. II–264.
[25] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.
[26] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 27, no. 10, pp. 1615–1630, 2005.
[27] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.
[28] M. Oquab, L. Bottou, I. Laptev, and J. Sivic, "Learning and transferring mid-level image representations using convolutional neural networks," in *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*. IEEE, 2014, pp. 1717–1724.
[29] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva, "Learning deep features for scene recognition using places database," in *Advances in Neural Information Processing Systems*, 2014, pp. 487–495.
[30] R. Girshick, F. Iandola, T. Darrell, and J. Malik, "Deformable part models are convolutional neural networks," *arXiv preprint arXiv:1409.5403*, 2014.

[31] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.
[32] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE, 2009, pp. 248–255.
[33] N. Zhang, J. Donahue, R. Girshick, and T. Darrell, "Part-based r-cnns for fine-grained category detection," in *Computer Vision–ECCV 2014*. Springer, 2014, pp. 834–849.
[34] 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 2010-21st British Machine Vision Conference*, 2010.
[35] G. Sharma, F. Jurie, and C. Schmid, "Discriminative spatial saliency for image classification," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, 2012, pp. 3506–3513.
[36] F. S. Khan, R. M. Anwer, J. van de Weijer, A. D. Bagdanov, A. M. Lopez, and M. Felsberg, "Coloring action recognition in still images," *International journal of computer vision*, vol. 105, no. 3, pp. 205–221, 2013.
[37] A. Prest, C. Schmid, and V. Ferrari, "Weakly supervised learning of interactions between humans and objects," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 34, no. 3, pp. 601–614, 2012.
[38] F. S. Khan, J. Xu, J. van de Weijer, A. D. Bagdanov, R. M. Anwer, and A. M. Lopez, "Recognizing actions through action-specific person detection," *Submitted for review*, 2015.
[39] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, "Liblinear: A library for large linear classification," *The Journal of Machine Learning Research*, vol. 9, pp. 1871–1874, 2008.
[40] C.-C. Chang and C.-J. Lin, "Libsvm: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 2, no. 3, p. 27, 2011.
[41] A. Oliva and A. Torralba, "Modeling the shape of the scene: A holistic representation of the spatial envelope," *International journal of computer vision*, vol. 42, no. 3, pp. 145–175, 2001.
[42] J. Wu and J. M. Rehg, "Centrist: A visual descriptor for scene categorization," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 33, no. 8, pp. 1489–1501, 2011.
[43] T. Dean, M. Ruzon, M. Segal, J. Shlens, S. Vijayanarasimhan, J. Yagnik *et al.*, "Fast, accurate detection of 100,000 object classes on a single machine," in *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*. IEEE, 2013, pp. 1814–1821.
[44] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," in *Computer Vision–ECCV 2014*. Springer, 2014, pp. 346–361.
[45] K. Grauman, F. Sha, and S. J. Hwang, "Learning a tree of metrics with disjoint visual features," in *Advances in neural information processing systems*, 2011, pp. 621–629.
[46] R. B. Girshick, P. F. Felzenszwalb, and D. McAllester, "Discriminatively trained deformable part models, release 5," http://people.cs.uchicago.edu/ rbg/latent-release5/.
[47] J. R. Uijlings, K. E. van de Sande, T. Gevers, and A. W. Smeulders, "Selective search for object recognition," *International journal of computer vision*, vol. 104, no. 2, pp. 154–171, 2013.