Convolutional Channel Features For Pedestrian, Face and Edge Detection

Bin Yang  Junjie Yan  Zhen Lei  Stan Z. Li
Center for Biometrics and Security Research & National Laboratory of Pattern Recognition
Institution of Automation, Chinese Academy of Sciences, China
{yb.derek, yanjjie}@gmail.com  {zlei, szli}@nlpr.ia.ac.cn

Abstract

In this paper, we revisit the multiple channel features approach proposed by Dollár et al. [10,11], which has shown excellent performances in various computer vision tasks. Enlightened by the ConvNets, we introduce an extended version of multiple channel features called Convolutional Channel Features (CCF), which transfers low-level features from off-the-shelf ConvNet models to feed the boosting classifiers based on decision trees. With the combination of CNN features and decision trees, CCF benefits from the rich capacity, robustness and sparsity in feature representation, as well as more efficiency in computation and storage during inference and learning process. Similar to multiple channel features, CCF is capable to solve diverse vision problems in a sliding window manner, and the computation cost in CCF multi-scale feature pyramid construction can be further reduced with power law based approximation in nearby scales and patchwork for shared convolution. We investigate into a large design space of CCF and show with experiments that CCF achieves leading performances in pedestrian detection, face detection, edge detection and object proposal generation. Codes are available at https://bitbucket.org/binyangderek/ccf.

1. Introduction

Many object detection solutions can be taken as the combinations of the feature extractor and classifier. Combinations such as Haar-like feature with boosting [40], HOG feature with SVM [8], multiple channel features with boosting [10] and fine-tuned high-level CNN features with SVM [17] have largely improved the object detection. Among these combinations, the multiple channel features with boosting and the fine-tuned high-level CNN features with SVM [17] show the most promising performances on various detection tasks.

The multiple channel features combined with boosting can be seen as an improved version of the Viola-Jones framework [40] with carefully hand-crafted channel feature representation and more sophisticated boosting algorithm. Multiple channel features showed great performance in pedestrian detection [11] at first, and was later generalized to face detection [41], edge detection [14] and object proposal generation [46]. Besides the accuracy, it typically runs at the frame rate speed and has very few parameters. Current bottleneck in their performance mainly lies in the representation capacity of the hand-crafted representation (LUV+HOG based) and therefore the performance rely largely on the implementation strategies.

The fine-tuned high-level CNN feature with SVM has recently shown extreme power [17, 32] in challenging tasks. Typically, a CNN model previously trained on ImageNet classification task is fine-tuned for new task and then the mid-level or high-level features of the CNN are extracted and fed into a SVM classifier. Owing to its large improvements in image classification, the learned feature hierarchies also set up new records in other vision tasks, like object detection [17], semantic segmentation [20] and fine-grained category detection [42]. The main advantage of this kind of approach is that the CNN has large capacity to handle large-scale training data and the learning process is end-to-end. However, currently CNN is often accompanied by huge computation complexity in inference and learning, and the model size is usually large (e.g., more than 500M for widely used VGG net [36]).

In detection, we always desire for better performance and lower computation/storage cost. This motivates us to build a bridge between the above two approaches and gain benefits from their respective advantages at the same time. Specifically, we extend the multiple channel features to low-level feature maps transferred from a CNN model trained on ImageNet image classification task, and replace the high-level connections in CNN with boosting learning based on decision trees. The advantages are two-fold: the transferred feature maps from CNN improve the representative capacity in channel features, while the boosting learning absolves the painstaking fine-tuning of high-level connections in CNN during the adaptation to various classification/regression vision problems. We name the new feature representation...
as Convolutional Channel Features (CCF). We explore the design choices concerning the CCF, give insights into the feature hierarchies learned in CNN and prove the representativeness and discriminativeness of low-level features in CNN. We also analyze several properties of CCF like feature maps approximation at nearby pyramid scales and shared convolution through patchwork [15, 22, 34], making CCF more computationally efficient.

We validate the CCF in various tasks. In classification problem, we follow the sliding-window mechanism to handle object detection with fixed aspect ratio (pedestrians and faces). It achieves the state-of-the-art performance for pedestrian and face detection. In regression problem, better performance is shown in edge detection than traditional channel features, and based on this CCF also applies to object proposal generation following the EdgeBox algorithm [46].

Our contributions are summarized as follows.

1. We extend the traditional hand-crafted multiple channel features to low-level feature representation transferred from CNN, namely CCF. CCF benefits from rich representative capacity of CNN and efficiency of multiple channel features. Motivated from early works [10, 14, 41, 46], we show that the CCF achieves state-of-the-art performances for pedestrian detection, face detection, edge detection and proposal generation in a sliding window manner. We release and maintain our codes at https://bitbucket.org/binyangderek/ccf.

2. Compared with many recent works [3, 31, 44] which accumulate more channel features in the hand-crafted manner, we show that the convolutions in low layers in CNN can naturally serve as the channels and the performance improves with the capacity of the CNN. It shows that we can get better channel features by just waiting for more powerful CNNs.

3. Different from recent works which use the high-level or mid-level features in CNN for representation, we show that the low-level representation also provides useful information when it is in conjunction with non-linear classifiers such as the boosting with decision trees. It provides a much more efficient way to substitute the “deep” layers of CNN in both inference and learning, leading to a model with much fewer parameters (typically fewer than 5M). Efficiency in computation and storage makes it possible that our model can be easily stored and used for mobile and embedded computing.

The remainder of the paper is organized as follows. The related work is reviewed in Sec. 2. In Sec. 3 we present the proposed convolutional channel features along with the design choices, computing techniques and model learning mechanism. We give analysis of the features and experiments on various tasks in Sec. 4. The paper is finally concluded in Sec. 5.

2. Related work

This work relates closely with two fundamental computer vision models, the framework of channel features combined with boosted trees [11], and the Deep Convolutional Neural Networks [26, 27]. The channel features approach, as one of the most influential subsequences of the seminal Viola and Jones framework [40], has been earning its reputation with the outstandingly effective and efficient LUV+HOG channel-wise feature representation, since the successful debut in pedestrian detection. The insight gained from the channel features is that, instead of letting machine learning algorithms look directly at the raw image pixels, we transform the image to a more algorithm-friendly form, which is the LUV+HOG 10-channel maps. Given that feature representation, a decision tree model learned in a boosting manner can achieve state-of-the-art performance in visual recognition task. Another important advantage of the channel features approach is that it is very efficient, as the processes of both feature extraction and model inference cost little computation. Besides visual recognition task (e.g. pedestrian detection), channel features are also successfully applied to other vision tasks like pose estimation [12], edge detection [14] and object proposal generation [47], further showing the generalization ability of the LUV+HOG feature representation. Recently, many followers of channel features have made a lot of efforts in improving it, classic variants like Roerei [3], LDCF [31], SquaresChnFtrs [3], InformedHaar [43] and Checkerboards [44] are proposed successively. One common contribution of these works, as pointed out in [44], is to apply hand-crafted filters on the LUV+HOG channels to further increase the representative capacity and robustness of the features. The hand-crafted filters can be in varying forms, like simply sum-pooling [3], unsupervised PCA filters [31], or human-designed Haar-like filters [43]. In an investigation of recent progress in pedestrian detection [4], it is pointed out that recent improvements in pedestrian detection are mainly from the better feature representation.

In the other camp, the Deep Convolutional Neural Networks have brought about a revolution in the computer vision community in recent years. Two components of CNN models play the key role in its success: 1) The feature representation is formed and learned in a hierarchical way, making it more representative. With increasing depth of layers, the representative capacity is enhanced, making it capable to handle recognition tasks of hundreds or thousands of categories. 2) CNN features have a good generalization ability. Specifically, [32] shows that mid-level CNN features can be transferred to a different dataset and achieves excellent performance after the fine-tuning process. With this technique, CNN features learned on large-scale recognition tasks have been successfully transferred to various vision tasks like object detection [17], semantic segmentation [28] and action
 detection [19]. The intuition under fine-tuning is that the large-scale learned CNN features at low and mid levels are very good over-complete representations of natural images and therefore are general to various vision tasks.

As the above two models are developed individually, we want to build a bridge between them, by replacing the hand-crafted channel features with low-level CNN features. There have already been some works that try to connect CNN models with popular vision models, like DPM [18,35] and RegionNet [48]. They both use the output of the last convolutional layer as feature representation and concatenate the CNN features with another structural model. There is also work that harnesses low-level CNN features, like [24] that uses the output of first convolutional layer to learn a two-class object classifier for proposal generation, then the RCNN approach is used for object detection. Our work differs from these in that 1) we use low-level CNN features as a general representation for visual problems ranging from low-level edge detection to high-level object detection; 2) we apply decision tree based boosting model directly on the feature maps to solve tasks like classification, regression and structural learning, which owns the advantages of fewer parameters, smaller model size and faster inference.

3. Proposed method

Based on the above motivation, we decide to extend the hand-crafted channel features to convolutional channel features, by transferring the image filters in CNN models to the feature extraction process of channel features. CCF has two main differences from the classic channel features approach: 1) The filters are learned rather than hand-crafted; 2) The filters are applied directly on raw images, rather than LUV+HOG channels. In the following part of this section, we will introduce the selection of CNN features used for CCF, techniques to accelerate the feature extraction process and how to learn decision tree models for different vision tasks. An investigative analysis of the CCF feature representation, and experimental results on pedestrian and face detection, edge detection and proposal generation will be demonstrated in the next section.

3.1. Selection of CNN features

We compare various feature choices among several popular CNN models, which are AlexNet(a.k.a. ANet) [26], VGG net [36] and GoogLeNet(a.k.a. GNet) [37]. The performance is evaluated through a standardized evaluation protocol that we personally defined on Caltech pedestrian benchmark [13]. Specifically, we use set00-04 as training set and set05 as test set. All images are sampled at an interval of 20 frames for the sake of evaluation efficiency. For fairness in comparison, we train an Aggregate Channel Features (ACF [10]) model and a Locally Decorrelated Channel Features (LDCF [31]) model as two baselines using the open source [9] toolbox. We adopt hard negative mining strategy in training baseline models, and all collected negative samples by ACF model is stored for all experiments on CNN feature selection. The model parameters for each case are fixed as well. Specifically, the window size is set to 128 × 64, and 2048 depth-3 decision trees are learned with RealBoost algorithm. There’s one exception which is the down-sampling factor. In multiple channel features, it has been proven [11] that a down-sampling factor of 4 performs best. In our cases, as different layers in CNN often have different number of pooling layers under them, we set a minimum down-sampling of 4 for all cases. That is to say, if the pooling factor of the extracted layer is smaller than 4, we add additional average pooling to the feature maps (marked as ‘x2’, ‘x4’ in the ‘Shrink’ column in Table 1). The entry name with ‘-s1’ means that we change the convolution stride of the first convolutional layer to 1 for an appropriate size of receptive field, since now we are extracting feature representation on an image patch that is much smaller than the 224 × 224 whole image.

In Table 1, we present results of various feature selections. The performance is measured by Log-average Miss Rate (MR) under the reasonable setting [13] (pedestrians taller than 50 pixels). We choose models all from large ones trained on ImageNet dataset, as these models outperform

| Net     | Layer | Map | Filter | Shrink | MR(%) |
|---------|-------|-----|--------|--------|-------|
| ACF     | -     | 10  | 3      | 4      | 41.22 |
| LDCF    | -     | 40  | 7      | 4      | 38.66 |
| ANet-s1 | conv1 | 96  | 11     | 1x4    | 61.65 |
|         | conv2 | 256 | 5      | 2x2    | 51.52 |
|         | conv3 | 384 | 3      | 4      | 43.73 |
|         | conv4 | 384 | 3      | 4      | 48.37 |
|         | conv5 | 256 | 3      | 4      | 53.37 |
| VGG-16  | conv2-2 | 128 | 3     | 2x2    | 53.86 |
|         | conv3-3 | 256 | 3     | 4      | 31.28 |
|         | conv4-4 | 512 | 3     | 8      | 27.66 |
|         | conv5-4 | 512 | 3     | 16     | 51.52 |
| VGG-19  | conv2-2 | 128 | 3     | 2x2    | 51.25 |
|         | conv3-4 | 256 | 3     | 4      | 33.56 |
|         | conv4-4 | 512 | 3     | 8      | 30.17 |
|         | conv5-4 | 512 | 3     | 16     | 55.55 |
| GNet    | conv2 | 192 | 3     | 4      | 45.06 |
|         | icp1  | 256 | -     | 8      | 38.44 |
|         | icp2  | 480 | -     | 8      | 31.66 |
|         | icp3  | 512 | -     | 16     | 35.99 |
| GNet-s1 | conv2 | 192 | 3     | 2x2    | 49.39 |
|         | icp1  | 256 | -     | 4      | 41.85 |
|         | icp2  | 480 | -     | 4      | 32.18 |
|         | icp3  | 512 | -     | 8      | 32.87 |

Table 1. Comparison of different feature choices evaluated on a small train/test split of Caltech training set.
small models trained on small datasets remarkably in our preliminary experiments. These models also have the advantage of simple image pre-processing which is just mean extraction. We bold the best performance in each entry, among which a $\sim 10\%$ improvement over the ACF and LDCF baselines can be seen. We first clarify the affect caused by changing the stride of first convolutional layer. By comparing between ‘GNet’ and ‘GNet-s1’, the change of stride leads to a little drop in performance. Based on this, our first observation is that large filter size acts poorly in channel features framework, by taking ‘ANet’ as an example. One possible reason is that larger filter size loses focus on local cues like edges, making it more appropriate for representing whole images rather than small patches. This observation also stands in ICF [11], where large local scale diminishes the performance greatly. The second observation is that the best choice in each entry is similar, i.e., around the convolutional layer whose down-sampling factor is 4 or 8. Through further experiments on more training data, we validate that ‘VGG-16 conv3-3’ performs best among all choices. There are two factors effecting this observation, the layer depth and the down-sampling factor. Since channel features are general-purposed and dense feature representation, it is reasonable to prefer low-level features in CNN to the task-specific, sparse high-level one (for a illustration of sparsity see Fig. 2.a).

3.2. Acceleration in feature extraction

Channel features are translation-invariant, so are convolutions, which makes it possible to deploy the sliding window approach in tackling with many different vision problems. With millions of weights, CNN is more computationally expensive compared with channel features. Fact is that we have already saved a large fraction of computation time by using low-level convolutional features. In addition to that, we try to bring more efficiency to the feature pyramid construction process because it is the speed bottleneck during testing. Below we introduce two techniques for efficient computation of multi-scale feature pyramid. Firstly we investigate the power law in feature scales to facilitate the approximation of feature maps at nearby scales in feature pyramid and prove that it holds on certain conditions. Secondly we demonstrate a popular skill used in multi-scale feature pyramid construction called patchwork. With the above two techniques combined, we observe a $5 \sim 10\times$ speedup in CCF pyramid computation.

**Power law in scales:** One significant reason that channel features are very efficient in sliding window style testing is the power law [10] used in feature approximation. The definition of power law in scales is that feature responses of specific feature types on natural images at different scales are subject to the power law. Therefore when computing the feature pyramid of an image, we can use the feature representation at one scale to approximate the feature representations at nearby scales instead of iterative computation at each scale. As the power law has been proved in LUV+HOG channel types, we wonder whether it still holds

![Figure 1. Power law fitting results of CCF at different scales on 100 randomly selected images.](image1.png)

![Figure 2. Illustration of feature sparsity and patchwork.](image2.png)
in the circumstance of CCF. Fig. 1 illustrates the power law fitting results of two CCF choices (VGG-16 conv3-3 and conv4-3) on two domains (faces and pedestrians). We determine whether power law holds based on the fitting deviation at nearby scales (i.e. small $\sigma$ at small scale values). In the domain of faces where images are of high-resolution and collected from the Internet, power law exists in conv3-3 (with $\sigma$ near 0 when $log_2(scale)$ is less than $-0.5$) but fails in conv4-4 (with $\sigma$ near 0.1 when $log_2(scale)$ is less than $-0.5$), as the latter feature representation is more sparse than the former. Experimental results in Sec. 4 further prove that power law holds in faces domain using conv3-3 features with negligible performance reduce. In the domain of pedestrians where VGA images are captured with a video camera, power law narrowly fits both feature representation (with large $\sigma$) and we observe an $\sim 3\%$ drop in performance on the above-mentioned evaluation protocol. The reason for the failure of power law in pedestrian images is likely to be the degraded image quality. Despite of the above facts, we believe that this founding is of value since often we are focusing on consumer images more and meanwhile the image quality from surveillance applications will become better and better.

**Patchwork:** Patchwork technique has been proposed early in [15] to accelerate the deformable part model. Recent works on CNN framework also borrow the same idea [22, 34]. As the input image size of a CNN model is usually fixed due to implementation restrictions, the idea of patchwork is straightforward that images at different scales can be put together to form a large input image for feature extraction (as shown in Fig. 2.b). The input image size of the CNN model is therefore set relatively large (say $1.5 \times$ of average image size). Patchwork can get considerable speed boost in feature pyramid construction especially at dense scales with no approximation, only if we handle the border effect between adjacent patches well. In our implementation, we add 16 pixel padding around each image when forming a large input image since the down-sampling factor of our CCF is 4. If the image is larger than the CNN input image size, we segment the image into small regions to fit into CNN input size rather than warp it. In doing so, our implementation can do patchwork with no approximation. Through experiments the patchwork can achieve $2 \times \sim 5 \times$ speed up compared with per-scale computation. The speed up factor depends on the implementation details and number of scales in feature pyramid. The source codes for efficient CCF extraction including both approximation at nearby scales according to power law and patchwork technique are released on the project website.

### 3.3. Model learning

After handling with the feature extraction and fast computation, we now move to the inference and learning part. Generally we follow the learning pipelines used in multiple channel features framework [10]. The learning workflow can be summarized as that ensemble of decision trees is learned on the candidate features formed by pixel values in CCF in a boosting manner. It is noted that for different tasks, the candidate features can have different forms, like single pixel lookups for pedestrian detection and pair-wise difference of two pixels for edge detection. While in inference stage, decision tree model is applied on dense image patches and output of each decision tree is accumulated to get the final result. Only two operations are needed during inference, which are pixel lookup and value comparison. Compared with fully connected layers in CNN model for inference, the decision tree model has the following superiorities: 1) It is light in model size, with only tens of thousands parameters compared with tens of millions in CNN models. 2) It is simple in learning, with off-the-shelf boosting algorithms available. 3) It is rich in capacity, with ensemble of non-linear weak classifiers and therefore few efforts in dealing with over-fitting or adversarial examples [38]. 4) It is universal in application, as the output of the leaf node can be either label, value or structure. 5) It is fast in inference, with only simple operations on values rather than matrix. In the following part we introduce in detail the models we use in different tasks.

**Pedestrian and face detection** Aggregate channel features have achieved state-of-the-art performances on pedestrian [10] and face [41] detection, whose learning pipeline is also used here. As the extracted convolutional feature maps are already down-sampled by a factor of 4 via max-pooling operation, no additional aggregation is needed and decision trees are learned directly on pixel lookups of the features. Since the feature representation is pooled through extraction, and has gone through several ReLU non-linear transformations, it is both sparse and informative (see Fig. 2.a for a visualization). Therefore no pre- or post-smoothing is needed. Randomness is beneficial during boosting training for better efficiency in time and memory as well as better generalization ability. Fig. 3 illustrates the visualization of pedestrian detectors learned with three different types of
channel features, showing that CCF is informative enough to encode structures and appearances. Quantitative experimental results are given in next section.

**Edge detection and proposal generation** For edge detection we deploy the structural learning approach in [14]. Model is built on $32 \times 32$ image patches, and candidate features are formed with pixel lookups and pairwise differences. We up-sample each image patch by a factor of 2 and then extract the convolutional feature maps of dimension $16 \times 16 \times 256$. Feature maps are smoothed with 2-pixel radius and 8-pixel radius each to generate pixel lookups and pairwise differences respectively. As the dimension of pairwise differences is very large even on a $32 \times 32$ model size, the 8 pixel radius smoothed feature maps are further downsampled to $5 \times 5$. In total, the number of candidate features is $16 \times 16 \times 256 + 300 \times 256 = 142,336$. The learning paradigm keeps the same as [14]. As for the proposal generation task, we inherit the algorithm introduced in [46], which is based on the results of edge detection.

4. Experiments

One reason that we select the low-level convolutional features is that it is a good balance between feature representativeness and generalization ability. Low-level features are just not so abstract to become task-specific, nor so basic to be sensitive to appearance variations. We verify this on four different tasks, ranging from low-level to high-level vision, which are pedestrian and face detection, edge detection and object proposal generation. On the benchmark of each task, the proposed approach achieves state-of-the-art performances and outperforms hand-crafted LUV+HOG channel features.

4.1. Implementation details

We choose ‘conv3-3’ layer’s output in VGG-16 model as our convolutional features. We implement feature extraction of CCF with Caffe toolkit [23]. Sliding window approach is deployed for all tasks during testing. Patchwork is used in CCF pyramid construction, where we use an input image size of $932 \times 932$ and set 6 scales per octave. Power law based approximation in nearby scales is used in face detection only (see Sec. 3.2 for reasons).

4.2. Feature analysis

Typically, in large CNN models, the number of feature maps is several hundreds, which is about tens of times more than that in channel features (except for filtered channel features which sometimes have thousands of maps). Therefore we want to know how these many feature maps work to boost the performance, or, in other words, what’s the performance like if we reduce the number of feature maps by selecting a fraction from all. Recall that the boosting process not only learns the model, but also plays the role of feature selection. So here we use an imperfect but straightforward way to determine how much significance each feature map accounts for in the decision forest model, which is to deploy the node split information in all learned decision trees. Specifically, for a learned decision forest model, we go through every node split in it and find out which feature map it belongs to. After checking all nodes, we get a distribution of occurrence of each feature map in the model. It can be conjectured that feature maps with higher occurrence in the model are of higher discriminativeness. Therefore we re-rank all feature maps and choose only the top 10, 40 and 128 maps as new CCFs to train new models. Before we go to the performance comparison part, let’s first see what’s the distribution of discriminativeness of all feature maps like. From Fig. 4.a we can see clearly different types of occurrence distribution with regard to different numbers of feature maps. For #maps = 10, the distribution is roughly uniform. When #maps goes up to 40, a small proportion of maps become dominant. As #maps continues to go up, the distribution has the trend of low-tail distribution. Taking the evaluation performance into consideration, there seems to be a positive correlation between the strength of dominance owned by a small fraction and the final performance.

In Fig. 4.b we compare CCF-10~CCF-256 with other methods in channel features family. A better feature representation should be closer to the bottom-left point in the figure. We can see that despite of worse performance of CCF-10 and CCF-40 compared with hand-crafted channel features, when #maps are larger than 100, CCF’s performance boosts by a large margin, while aggressively adding feature maps (filtered channel features) gets limited improvements. This shows the superiority of convolutional feature representation over hand-crafted one, as CCF provides a better over-complete representation of natural images. The reason for the failure in small #maps may be that greedily selected discriminative maps doesn’t guarantee a better representation when used as a whole due to existence of intra-correlations between channel maps.
4.3. Pedestrian detection

Caltech benchmark [13] is used for training and testing. Positive data is collected at a sampling interval of 4 frames, while negative samples are collected by training a baseline Aggregate Channel Features (ACF) detector with 3 rounds of hard negative mining. We set model size to $128 \times 64$ for feature extraction and up-sample the image with a factor of 2 in feature pyramid construction for detecting pedestrians taller than 50 pixels. The evaluation results are shown in Fig. 5.a. We set the new record of methods using CNN features on the benchmark with 18.71% Log-average Miss Rate, outperforming previous CNN based methods Joint-Deep [33], SDN [29], AlexNet+ImageNet [21] by a large margin. Compared with methods from the channel features family, the CCF has great advantages over aggregate channel features approaches ACF [10] and LDCF [31], and is also on par with the carefully designed Checkerboards [44] which uses more feature maps (610 VS 256) and more training data (3Hz VS 4Hz). When adding 10 ACF channels to the CCF, we get an additional 1.4% improvement and set the new record of algorithms without using temporal information. Note that our approach shows good performance at low FPPIs, while CCF and CCF+CF ranks first at 0.01 and 0.1 FPPI respectively.

4.4. Face detection

Since there doesn’t exist a well-designed face detection benchmark, it’s hard to say which part of the detector contributes most to the final performance, when comparing different algorithms on the same test set. As we are focusing more on the feature representation part, we adopt the ACF approach [41] for face detection as a strong baseline. We use the same training data as [41] and similar boosting paradigm (the depth of decision tree is changed from 2 to 3). An interesting observation is that with CCF feature representation, we now can handle multiple views with one boosting model instead of training one model for each view as in [41]. We evaluate the CCF face detector on the AFW face detection test set [45], which contains 205 high-resolution Internet images in unconstrained setting. We use the evaluation toolbox provided by [30], as it solves the problems of missing annotations and bounding box adjustment to some extent. From the evaluation results shown in Fig. 5.b, CCF surpasses the baseline ACF in AP value by a $\sim 1.5\%$ improvement, while is competitive with state-of-the-arts DPM [30] and HeadHunter [30] which deploy multi-view models. What’s more, CCF achieves the highest maximum recall rate among all algorithms. We also verify the power law feature approximation technique (described in Sec. 3.2) in face detection for acceleration (entry ‘Our CCF-Approx’ in the figure). In average we observe a $\sim 2.4 \times$ speed up during testing on AFW test set with patchwork deployed and $\sim 4 \times$ speed up without patchwork. From the evaluation results, the performance degradation is tolerable compared with the time saved.

4.5. Edge detection

We deploy the structured forests approach SE [14] for edge detection. For the feature representation part, as the feature down-sampling factor in SE is 2, while CCF has a down-sampling factor of 4, we up-sample the input image by a factor of 2. Also, SE uses multi-scale gradient channels, while we observe no improvements in using multi-scale CCF, therefore we just use the single-scale one. We train and evaluate the algorithm on BSDS500 dataset. From the results in Table 2, we can see that CCF outperforms SE with single-scale detection in all three metrics. When multi-scale detection (marked as ‘ms’ in the table) is used, SE gets a considerable improvement while CCF improves only.
a little. As for comparison with other state-of-the-art algorithms, it seems that different algorithms have advantages with respect to different metrics. Particularly, CCF beats other state-of-the-arts in AP value. What’s more, adding 10 LUV+HOG channels to CCF feature representation, like what we do in pedestrian detection, can improve the ODS and OIS metrics a little but degrades the AP value.

4.6 Object proposal generation

Since CCF shows competitive performances in edge detection, we adopt the EdgeBoxes approach [46], which generates object proposals directly from edges, for proposal generation task. We replace the edge detection module with the one described above. For time efficiency, we use a single-scale version of CCF based edge detector. We evaluate the CCF based algorithm on the PASCAL VOC 2007 test set [16], which has well-annotated object bounding boxes. The results are shown in Table 3. Compared with other state-of-the-art algorithms, CCF achieves the best result in all four metrics, which are Area Under Curve (AUC), number of proposals needed to reach 50% and 75% recall and maximum recall rate.

4.7 Speed

As CCF achieves state-of-the-art performances in various tasks, the main concern on its practical application seems to be the large computation cost caused by high dimensional hierarchical features. However, we have made some efforts for the efficiency in both training and testing. On one hand, when compared with conventional channel features approach, as we use a more complicated feature representation, the time cost of feature extraction increases. However, we verify the power law property of CCF in Internet images (Sec. 3.2), and validate this finding on face detection task (Sec. 4.4). The inherence of power law property compensates for the additional computation time. Also, during boosting training, we find the randomness in features selected by each decision tree node can provide less computation and better generalization performance. On the other hand, in comparison with CNN based methods, we adopt the patchwork technique to efficiently compute multi-scale CNN feature pyramid (Sec. 3.2), and transfer only low-level features from a large CNN model (Sec. 3.1), leading to faster feature extraction than traditional ConvNet methods.

5. Conclusion

In this paper, we revisit the popular multiple channel features and extend it to convolutional channel features (CCF), which benefits from the rich representative capacity of CNN to guarantee the outstanding performance in various vision tasks, as well as the efficiency in inference and learning from boosting decision tree model. Additionally, the model size is very small (fewer than 5M parameters), which enables it to be easily used in mobile and embedded computing task. We validate different settings of the CCF and give some insights to get satisfied performance. We also provide practical techniques such as patchwork and approximation of nearby scales to accelerate the computation. In conjunction with decision forest model, we show that the proposed CCF approach achieves state-of-the-art performances for diverse tasks ranging from pedestrian and face detection to edge detection and object proposal generation.

References

[1] B. Alexe, T. Deselaers, and V. Ferrari. What is an object? In Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on, pages 73–80. IEEE, 2010.
[2] P. Arbelaez, J. Pont-Tuset, J. Barron, F. Marques, and J. Malik. Multiscale combinatorial grouping. In Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on, pages 328–335. IEEE, 2014.
[3] R. Benenson, M. Mathias, T. Tuytelaars, and L. Van Gool. Seeking the strongest rigid detector. In CVPR. IEEE, 2013.
[4] R. Benenson, M. Omran, J. Hosang, , and B. Schiele. Ten years of pedestrian detection, what have we learned? In ECCV, CVRSUAD workshop, 2014.
[5] G. Bertasius, J. Shi, and L. Torresani. Deepedge: A multi-scale bifurcated deep network for top-down contour detection. arXiv preprint arXiv:1412.1123, 2014.
[6] J. Carreira and C. Sminchisescu. Constrained parametric min-cuts for automatic object segmentation. In Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on, pages 328–335. IEEE, 2014.

Table 2. Evaluation results of edge detection on BSDS500 dataset.

| Method       | ODS | OIS | AP  |
|--------------|-----|-----|-----|
| Human        | 0.80| 0.80| -   |
| DeepNet [25] | 0.738| 0.759| 0.758 |
| SE [14]      | 0.739| 0.759| 0.792 |
| SE+ms [14]   | 0.746| 0.767| 0.803 |
| MCG [2]      | 0.747| 0.779| 0.759 |
| DeepEdge [5] | 0.753| 0.772| 0.807 |
| CCF          | 0.741| 0.761| 0.808 |
| CCF+ms       | 0.744| 0.767| 0.809 |
| CCF+ms+CF    | 0.745| 0.768| 0.807 |

Table 3. Evaluation results of object proposal generation on PASCAL VOC 2007 test set with IoU threshold of 0.7. Metrics are Area Under Curve (AUC), number of proposals needed to reach 50% and 75% recall and maximum recall rate.

| Methods    | AUC | N@50% | N@75% | Recall |
|------------|-----|-------|-------|--------|
| BING [7]   | 0.20| -     | -     | 29%    |
| Objectness [1] | 0.27| 584   | -     | 68%    |
| Sel. Search [39] | 0.40| 199   | 1434  | 87%    |
| CPMC [5]   | 0.41| 111   | -     | 65%    |
| EdgeBoxes [46] | 0.46| 108   | 800   | 87%    |
| CCF        | 0.48| 89    | 649   | 88%    |
Recognition (CVPR), 2010 IEEE Conference on, pages 3241–3248. IEEE, 2010. 8

[7] M.-M. Cheng, Z. Zhang, W.-Y. Lin, and P. Torr. Bing: Binarized normed gradients for objectness estimation at 300fps. In Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on, pages 3286–3293. IEEE, 2014. 8

[8] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In CVPR, IEEE, 2005. 1

[9] P. Dollár. Pietr’s Computer Vision Matlab Toolbox (PMT). http://vision.ucsd.edu/~pdollar/toolbox/doc/index.html. 3

[10] P. Dollár, R. Appel, S. Belongie, and P. Perona. Fast feature pyramids for object detection. PAMI, 2014. 1, 2, 3, 4, 5, 7

[11] P. Dollár, Z. Tu, P. Perona, and S. Belongie. Integral channel features. In BMVC, 2009. 1, 2, 3, 4

[12] P. Dollár, P. Welinder, and P. Perona. Cascaded pose regression. In CVPR, 2010. 2

[13] P. Dollár, C. Wojek, B. Schiele, and P. Perona. Pedestrian detection: An evaluation of the state of the art. PAMI, 34, 2012. 3, 7

[14] P. Dollár and C. Zitnick. Fast edge detection using structured forests. PAMI, 2014. 1, 2, 6, 7, 8

[15] C. Dubout and F. Fleuret. Exact acceleration of linear object detectors. In Computer Vision—ECCV 2012, pages 301–311. Springer, 2012. 2, 5

[16] 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, 88(2):303–338, 2010. 8

[17] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In CVPR, IEEE, 2014. 1, 2

[18] R. Girshick, F. Iandola, T. Darrell, and J. Malik. Deformable part models are convolutional neural networks. arXiv preprint arXiv:1409.5401, 2014. 3

[19] G. Gkioxari, B. Hariharan, R. Girshick, and J. Malik. R-cnn for pose estimation and action detection. arXiv preprint arXiv:1406.5212, 2014. 3

[20] S. Gupta, R. Girshick, P. Arbelaez, and J. Malik. Learning rich features from rgb-d images for object detection and segmentation. In ECCV, Springer, 2014. 1

[21] J. Hosang, M. Omran, R. Benenson, and B. Schiele. Taking a deeper look at pedestrians. In CVPR, 2015. 7

[22] F. Iandola, M. Moskewicz, S. Karayev, R. Girshick, T. Darrell, and K. Keutzer. Densenet: Implementing efficient convnet descriptor with dense channels. arXiv preprint arXiv:1404.1869, 2014. 2, 5

[23] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. In Proceedings of the ACM International Conference on Multimedia, pages 675–678. ACM, 2014. 6

[24] N. Karianakis, T. J. Fuchs, and S. Soatto. Boosting convolutional features for robust object proposals. arXiv preprint arXiv:1503.06510, 2015. 3

[25] J. J. Kivinen, C. K. Williams, N. Heess, and D. Technologies. Visual boundary prediction: A deep neural prediction network and quality dissection. In AISTATS, volume 1, page 9, 2014. 8

[26] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems, pages 1097–1105, 2012. 2, 3

[27] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11):2278–2324, 1998. 2

[28] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. arXiv preprint arXiv:1407.3767, 2014. 2

[29] P. Luo, Y. Tian, X. Wang, and X. Tang. Switchable deep network for pedestrian detection. In Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on, pages 899–906. IEEE, 2014. 7

[30] M. Mathias, R. Benenson, M. Pedersoli, and L. Van Gool. Face detection without bells and whistles. In Computer Vision—ECCV 2014, pages 720–735. Springer, 2014. 7

[31] W. Nam, P. Dollár, and J. H. Han. Local decorrelation for improved pedestrian detection. In Advances in Neural Information Processing Systems, pages 424–432. IEEE, 2014. 2, 3, 7

[32] 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, pages 1717–1724. IEEE, 2014. 1, 2

[33] W. Ouyang and X. Wang. Joint deep learning for pedestrian detection. In Computer Vision (ICCV), 2013 IEEE International Conference on, pages 2056–2063. IEEE, 2013. 7

[34] G. Papandreou, I. Kokkinos, and P.-A. Savalle. Untangling local and global deformations in deep convolutional networks for image classification and sliding window detection. arXiv preprint arXiv:1412.0296, 2014. 2, 5

[35] P.-A. Savalle, S. Tsokgas, G. Papandreou, and I. Kokkinos. Deformable part models with cnn features. In 3rd Parts and Attributes Workshop, ECCV, volume 8. 3

[36] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556, 2014. 1, 3

[37] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. arXiv preprint arXiv:1409.4842, 2014. 3

[38] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus. Intriguing properties of neural networks. arXiv preprint arXiv:1312.6199, 2013. 5

[39] J. R. Uijlings, K. E. van de Sande, T. Gevers, and A. W. Smeulders. Selective search for object recognition. International journal of computer vision, 104(2):154–171, 2013. 8

[40] P. Viola and M. J. Jones. Robust real-time face detection. IJCV, 2004. 1, 2

[41] B. Yang, J. Yan, Z. Lei, and S. Z. Li. Aggregate channel features for multi-view face detection. In IJCB. IEEE, 2014. 1, 2, 5, 7

[42] N. Zhang, J. Donahue, R. Girshick, and T. Darrell. Part-based r-cnns for fine-grained category detection. In ECCV. Springer, 2014. 1

[43] S. Zhang, C. Bauckhage, and A. B. Cremers. Informed haar-like features improve pedestrian detection. In Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on, pages 947–954. IEEE, 2014. 2

[44] S. Zhang, R. Benenson, and B. Schiele. Filtered channel features for object detection without bells and whistles. In Computer Vision—ECCV, 2014, pages 424–432, 2014. 2, 3

[45] X. Zhu and D. Ramanan. Face detection, pose estimation, and landmark localization in the wild. In Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, pages 2879–2886. IEEE, 2012. 7

[46] C. L. Zitnick and P. Dollár. Edge boxes: Locating object proposals from edges. In ECCV. Springer, 2014. 1, 2, 6, 8

[47] C. L. Zitnick and P. Dollár. Edge boxes: Locating object proposals from edges. In ECCV. Springer, 2014. 2

[48] W. Y. Zou, X. Wang, M. Sun, and Y. Lin. Generic object detection with dense neural patterns and regionlets. arXiv preprint arXiv:1404.4516, 2014. 3