Convolutional Features for Correlation Filter Based Visual Tracking

Martin Danelljan,[*] Gustav Häger,[*] Fahad Shahbaz Khan, Michael Felsberg

Computer Vision Laboratory, Linköping University, Sweden

{martin.danelljan, gustav.hager, fahad.khan, michael.felsberg}@liu.se

Abstract

Visual object tracking is a challenging computer vision problem with numerous real-world applications. This paper investigates the impact of convolutional features for the visual tracking problem. We propose to use activations from the convolutional layer of a CNN in discriminative correlation filter based tracking frameworks. These activations have several advantages compared to the standard deep features (fully connected layers). Firstly, they mitigate the need of task specific fine-tuning. Secondly, they contain structural information crucial for the tracking problem. Lastly, these activations have low dimensionality. We perform comprehensive experiments on three benchmark datasets: OTB, ALOV300++ and the recently introduced VOT2015. Surprisingly, different to image classification, our results suggest that activations from the first layer provide superior tracking performance compared to the deeper layers. Our results further show that the convolutional features provide improved results compared to standard hand-crafted features. Finally, results comparable to state-of-the-art trackers are obtained on all three benchmark datasets.

1. Introduction

Visual tracking is the task of estimating the trajectory of a target object in an image sequence. It has many important real-world applications, such as robotics [11] and road scene understanding [18]. In the generic tracking problem, the target can be any object, and only its initial location is known. This problem is challenging due to several factors, such as appearance changes, scale variations, deformations and occlusions. Most state-of-the-art approaches tackle the tracking problem by learning a discriminative appearance model of the target object. Such approaches [9, 12, 21] rely on rich feature representations for describing of the target and background appearance. This paper investigates robust feature representations for visual tracking.

Among the discriminative tracking methods, correlation filter based approaches have recently shown excellent performance on benchmark tracking datasets [24, 39]. These

---

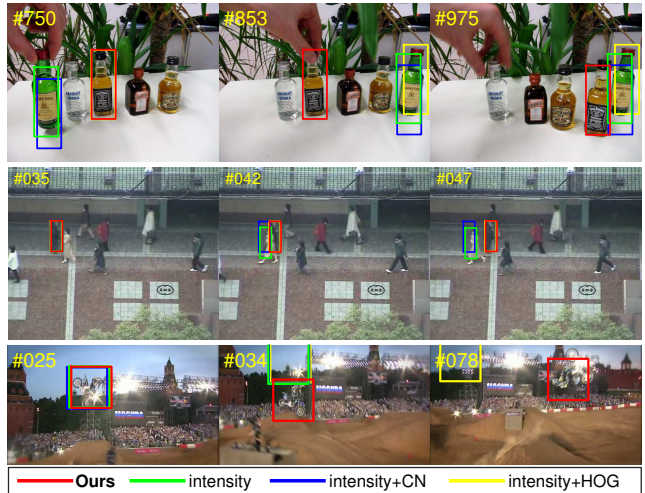[1] Both authors contributed equally to this work.

Figure 1. A comparison of the proposed feature representation with three commonly employed hand-crafted features, namely image intensity, Color Names (CN) and Histogram of Oriented Gradients (HOG). Tracking results from our DCF tracker on three example sequences are shown. The convolutional features used in our tracker provides a richer description of target apperance, leading to better performance.

approaches learn a discriminative correlation filter (DCF), from example patches of the target appearance. Initially, the DCF framework was restricted to a single feature channel (e.g. a grayscale image) [4]. Later works have investigated extending the single-channel DCF to using multi-channel feature representations for tracking [12]. However, existing DCF based approaches [12, 21, 4] suffer from the periodic boundary effects induced by circular correlation. Only recently, Danelljan et al. [10] proposed Spatially Regularized Discriminative Correlation Filters (SRDCF) to mitigate the negative effects of the inherent periodic assumption of the standard DCF. In this work, we investigate convolutional features within both the standard DCF framework and the more recent SRDCF framework.

Initially, most tracking approaches relied on using only image intensity information or simple color transformations [4, 30, 32] for feature representation. In recent years, hand-crafted histogram-based descriptors have shown improved results for visual tracking. Feature representations such as

HOG [21], Color Names [12] and channel representations [8] have successfully been employed in DCF based tracking frameworks. These descriptors aim at capturing the shape, color or luminance information of the target appearance. Combining multiple features have also been investigated [28] within a DCF framework.

Recently, Convolutional Neural Networks (CNNs) have significantly advanced the state-of-the-art in many vision applications, including object recognition [25, 31] and object detection [19]. These networks take a fixed sized RGB image as input to a sequence of convolution, local normalization and pooling operations (called layers). The final layers in the network are fully connected (FC), and are typically used to extract features for classification. CNNs require a large amount of training data, and are trained on the large scale ImageNet dataset [13]. It has been shown that the deep features extracted from the network (the FC layer) are generic and can be used for a variety of vision applications [2].

As discussed above, the common strategy is to extract deep features from the activations of the FC layer of the pre-trained network. Other than the FC layer, activations from convolutional layers of the network have recently been shown to achieve superior results for image classification [6]. These convolutional layers are discriminative, semantically meaningful and contain structural information crucial for the localization task. Additionally, the use of convolutional features mitigates the need of task-specific fine-tuning employed with standard deep features. In such approaches, it has been shown that activations from the last convolutional layer provides improved results compared to other layers of the same network [6].

In this work, we investigate the impact of convolutional features in two DCF based tracking frameworks: a standard DCF framework and the SRDCF framework [10]. Contrary to in image classification, we show that activations from the first layer provides superior tracking performance compared to the deeper layers of the network. Finally, we provide both qualitative and quantitative comparison of convolutional features with standard hand-crafted histogram descriptors, commonly used within the DCF based trackers.

Comprehensive experiments are performed on three benchmark datasets: the Online Tracking Benchmark (OTB) [39], the Amsterdam Library of Ordinary Videos for tracking (ALOV300++) [35] and the Visual Object Tracking (VOT) challenge 2015 [1]. Our results demonstrate that superior performance is obtained by using convolutional features compared to standard hand-crafted feature representations. Finally, we show that our proposed tracker achieves state-of-the-art tracking performance on all three benchmark datasets. Figure 1 provides a comparison of our tracker employing convolutional features with commonly used feature representations within the same DCF based tracking framework.

The paper is organized as follows. Section 2 discusses related work in tracking and convolutional neural networks. Our tracking framework is described in section 3. The employed DCF and SRDCF frameworks are briefly presented in section 3.1 and section 3.2 respectively, while the used convolutional features are discussed in section 3.3. Section 4 contains the experimental evaluations and results. Finally, conclusions are provided in section 5.

2. Related Work

The visual tracking problem can be approached using generative [34, 22] or discriminative [20, 3, 40] appearance models. The latter methods apply machine learning techniques to discriminate the target appearance from the background. Recently, the Discriminant Correlation Filter (DCF) [4] based approaches have achieved state-of-the-art results on benchmark tracking datasets [24, 39]. The success of DCF based methods is evident from the outcome of the Visual Object Tracking (VOT) 2014 challenge [24], where the top three entries employ variants of the DCF framework. Related methods [12, 21] have also shown excellent results on the Object Tracking Benchmark (OTB) [39]. In this work, we employ the DCF framework to investigate the impact of convolutional features for tracking.

The DCF based tracking approaches learn a correlation filter to discriminate between the target and background appearance. The training data is composed of observed samples of the target appearance and the surrounding background. Bolme et al. [4] initially proposed the MOSSE tracker, which is restricted to using a single feature channel, typically a grayscale image. Henriques et al. [21] introduced a kernelized version of the tracker, to allow non-linear classification boundaries. More recent work [12, 9, 21] have achieved significant increase in tracking performance by investigating the use of multi-dimensional features in the DCF tracking framework.

Despite their success, it is known that standard DCF based trackers greatly suffers from the periodic assumption induced by circular correlation. This leads to inaccurate and insufficient training samples as well as a restricted search area. Galoogahi et al. [16] propose to solve a constraint problem using the Alternating Direction Method of Multipliers (ADMM) to preserve the correct filter size. This method is however restricted to using a single feature channel and hence not applicable for our purpose. Recently, Danelljan et al. [10] tackles these issues by introducing the Spatially Regularized DCF (SRDCF). Their approach allows the expansion of the training and search regions without increasing the effective filter size. This increases the discriminative power and robustness of the tracker, leading to a significant performance gain. Moreover, the filter is optimized directly in the Fourier domain using Gauss-Seidel,

while every ADMM iteration in [16] requires a transition between the spatial and Fourier domain.

In the last few years, convolutional neural networks (CNN) have significantly advanced the state-of-the art in object recognition and detection benchmarks [33]. The CNNs learn invariant features by a series of convolution and pooling operations. These layers of convolution and pooling operations are followed by one or more fully connected (FC) layers. The entire CNNs are trained using raw pixels with a fixed input size. In order to train these networks, a large amount of labeled training data [26] is required. The activations of fully connected layers in a trained deep network are known to contain general-purpose features applicable to several visual recognition tasks such as attribute recognition, action recognition and scene classification [2].

Interestingly, recent results [6, 29] suggest that improved performance is obtained using convolutional layer activations instead of those extracted from the fully connected layers of the same network. The convolutional layers in deep networks are discriminative, semantically meaningful and mitigate the need to apply task specific fine-tuning. The work of [29] proposes a cross-convolutional layer pooling approach. The method works by employing feature maps of one convolutional layer as local features. The image representation is obtained by pooling the extracted features using the feature maps of the successive convolutional layers. A multi-scale convolutional feature based approach is proposed by [6] for texture classification and object recognition. In their method, activations from the convolutional layer of the pre-trained network are used as local features. Further, it was shown that the activations of the last convolutional layer of the network provide superior performance compared to other layers [6] for visual recognition.

Despite the success of deep features in several computer vision tasks, less attention has been dedicated to investigate deep features in the context of visual tracking. A human tracking algorithm is proposed by Fan et al. [14] by learning convolutional features from offline training data. The authors of [38] propose a compact deep feature based tracking framework that learns generic features by employing a stacked denoising auto-encoder. Zhou et al. [42] investigate boosting techniques to construct an ensemble of deep networks for visual tracking. Li et al. [27] propose a deep tracking framework using a candidate pool of multiple CNNs. Different from the above mentioned work, we investigate the impact of deep features for DCF based tracking. We exploit the spatial structure of the convolutional features for learning a DCF (or SRDCF), which acts as a final classification layer in the network. In this paper, we also investigate the performance of different convolutional layers and compare with standard hand-crafted features.

3. Method

Our tracking approach is based on learning a DCF or a SRDCF from samples of the target appearance. For image description, we employ convolutional features extracted from these samples. In each new frame, the learned DCF is applied on the convolutional features extracted from the predicted target location. A location estimate is then achieved by maximizing the detection scores.

3.1. Discriminative Correlation Filters

In this work, we use a standard DCF framework to investigate the impact of convolutional features for tracking. The DCF framework utilizes the properties of circular correlation to efficiently train and apply a classifier in a sliding window fashion. The resulting classifier is a correlation (or convolution) filter which is applied to the input feature channels. Hence, the correlation operation within the DCF acts similarly to a convolutional layer in a CNN. The corresponding learned filter can be viewed as a final convolutional classification layer in the network. Unlike the costly methods typically applied for training CNNs, the DCF is trained efficiently by solving a linear least-squares problem and exploiting the Fast Fourier Transform (FFT).

The discriminative correlation filer f_t is learned from a set of example patches x_k which are sampled at each frame $k = 1, \ldots, t$. Here, t denotes the current frame number. The patches are all of the same size and are typically centered at the estimated target location in each frame. We denote feature channel j of x_k by superscript x_k^j. In our case, x_k^j corresponds to the output of channel j at a convolutional layer in the CNN. The objective is to learn a correlation filter f_t^j for each channel j, that minimizes the following loss,

$$\epsilon = \sum_{k=1}^{t} \alpha_k \|f_t \star x_k - y_k\|^2 + \lambda \|f_t\|^2. \qquad (1)$$

Here $\star$ denotes circular correlation generalized to multi-channel signals in the conventional way by computing inner products. That is, the correlation output for each channel is summed over the channel dimension to produce a single-channel output. The desired correlation output y_k is set to a Gaussian function with the peak placed at the target center location [4]. A weight parameter λ controls the impact of the regularization term, while the weights α_k determine the impact of each training sample.

To find an approximate solution of (1), we use the online update rule of [9]. At frame t, the numerator $\hat{g}_t$ and denominator $\hat{h}_t$ of the discrete Fourier transformed (DFT) filter $\hat{f}_t$

are updated as,

$$\hat{g}_t^j = (1 - \gamma)\hat{g}_{t-1}^j + \gamma\overline{\hat{y}_t} \cdot \hat{x}_t^j \tag{2a}$$

$$\hat{h}_t = (1 - \gamma)\hat{h}_{t-1} + \gamma\left(\sum_{l=1}^{d}\overline{\hat{x}_t^l} \cdot \hat{x}_t^l + \lambda\right). \tag{2b}$$

Here, the hat denotes the 2-dimensional DFT, the bar denotes complex conjugation and $\cdot$ denotes pointwise multiplication. The scalar $\gamma \in [0, 1]$ is a learning rate parameter and d is the number of feature channels. The sought filter can then be constructed by a point-wise division $\hat{f}_t^j = \hat{g}_t^j/\hat{h}_t$.

To locate the target at frame t, a sample patch z_t is first extracted at the previous location. The filter is then applied by computing the correlation scores in the Fourier domain

$$s_t = \mathscr{F}^{-1}\left\{\sum_{j=1}^{d}\overline{\hat{f}_{t-1}^j} \cdot \hat{z}_t^l\right\}. \tag{3}$$

Here, $\mathscr{F}^{-1}$ denotes the inverse DFT. To obtain an estimate of the target scale, we apply the learned filter at multiple resolutions. The target location and scale in the image are then updated by finding the maximum correlation score over all evaluated locations and scales.

3.2. Spatially Regularized Discriminative Correlation Filters

As discussed above, the conventional DCF tracking approaches have demonstrated impressive performance in recent years. However, the standard DCF formulation is severely hampered by the periodic assumption introduced by the circular correlation. This leads to unwanted periodic boundary effects at both the training and detection stages. Such periodic boundary effects limit the performance of the DCF in several aspects. First, the DCF trackers struggle in cases of fast motion due to a restricted search region. More importantly, the inaccurate and insufficient training data limit the discriminative power of the learned model and lead to over-fitting.

To mitigate the periodic boundary effects, Danelljan et al. [10] recently proposed Spatially Regularized Correlation Filters (SRDCF), leading to a significant performance boost for correlation based trackers. The authors introduced a spatial regularization function w that penalizes filter coefficients residing outside the target bounding box. This allows an expansion of the training and detection regions without increasing the effective filter size. Instead of (1), the following cost is minimized,

$$\epsilon = \sum_{k=1}^{t}\alpha_k\|f_t \star x_k - y_k\|^2 + \sum_{l=1}^{d}\|w \cdot f_t^l\|^2. \tag{4}$$

The spatial regularization function w reflects the reliability of visual features depending on their spatial location. The function w is therefore set to smoothly increase with distance from the target center, as suggested in [10]. Since background coefficients in the filter f_t are suppressed by assigning larger weights in w, the emphasis on background information at the detection stage is reduced. On the contrary, a naive expansion of the sample size (using a standard regularization) would also result in a similar increase in the effective filter size. However, this leads to a large emphasis on background features, thereby severely degrading the discriminative power of the learned model.

The cost (4) can be efficiently minimized in the Fourier domain by exploiting the sparsity of the DFT coefficients $\hat{w}$. Instead of relying on approximate solutions, such as (2), [10] propose an iterative minimization scheme based on Gauss-Seidel, that converges to the global minimum of (4). We refer to [10] for a detailed description of the SRDCF training procedure.

3.3. Convolutional Features for DCF Tracking

Traditionally, DCF based approaches rely on hand-crafted features for image description [12, 21, 28]. In this work, we instead investigate the use of convolutional layer activations for DCF based tracking. We employ the imagenet-vgg-2048 network [5] using the implementation in the MatConvNet library [37].[1] The network is trained on the ImageNet dataset, for the image classification task. The employed network contains five convolutional layers and uses a 224×224 RGB image as an input. At each convolutional layer, we employ the activations produced after the rectified linear (ReLu) non-linearity. The samples used for training and detection in the DCF framework (x_k and z_k respectively) are obtained by extracting the convolutional features at the appropriate image location.

When computing the convolutional features, the image patch is pre-processed by first resizing it to the input size (224×224) and then subtracting the mean of the network training data. For grayscale images, we simply set the R, G and B-channels equal to the grayscale intensities. As discussed in [4], the extracted features are always multiplied with a Hann window.

4. Experiments

We perform experimental evaluation on three public benchmark datasets: the Online Tracking Benchmark (OTB) [39], the Amsterdam Library of Ordinary Videos for tracking (ALOV300++) [35] and the Visual Object Tracking (VOT) challenge 2015 [1].

---

[1]The network is available at http://www.vlfeat.org/matconvnet/models/imagenet-vgg-m-2048.mat
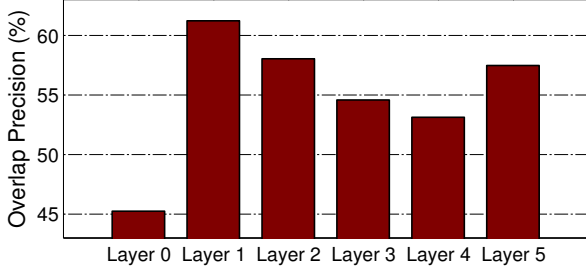
Figure 2. Comparison of tracking performance when using different convolutional layers in the network. The mean overlap precision over all color videos in the OTB dataset is displayed. The input RGB image (layer 0) provides inferior performance compared to the convolutional layers. The best results are obtained using the first convolutional layer. The performance then degrades for each deeper layer in the network, until the final layer.

	Layer 0	Layer 1	Layer 2	Layer 3	Layer 4	Layer 5
Spatial size	224×224	109×109	26×26	13×13	13×13	13×13
Dimensionality	3	96	256	512	512	512

Table 1. The spatial size and dimensionality of the convolutional features extracted from the employed network. Layer 0 denotes the input RGB image, after the necessary preprocessing steps.

4.1. Feature comparison

We start by evaluating the different convolutional layers of the imagenet-vgg-2048 network [5], as described in section 3.3. For simplicity, we employ the standard DCF framework described in section 3.1 but without any scale estimation, for this experiment. The evaluation is performed on all 35 color videos in the OTB dataset [39]. The results are presented in terms of overlap precision (OP). It is computed as the percentage of frames in a sequence where the intersection-over-union overlap with the ground-truth bounding box is larger than a threshold $T \in [0, 1]$. In tables and figures, we report the overlap precision at a threshold of $T = 0.5$, which corresponds to the PASCAL criterion. We also provide more detailed results in the *success plots*, where OP is plotted over the range of thresholds. In this case we use the *area-under-the-curve* (AUC) to rank the different methods. The AUC is displayed in the legend for each tracker. For more details regarding the evaluation protocol, we refer to [39].

Figure 2 shows the mean overlap precision, at the threshold $T = 0.5$, of the input layer (layer 0) and the five convolutional layers in the network. All convolutional layers significantly outperform the input layer, consisting of a resized and normalized RGB image. Unlike image classification, the first convolutional layer achieves the best tracking results. The performance then drops for each deeper layer in the network, until the final layer. We partly attribute this effect to the decreased spatial resolution in the deeper layers (see table 1). Intuitively, better spatial resolution alleviates
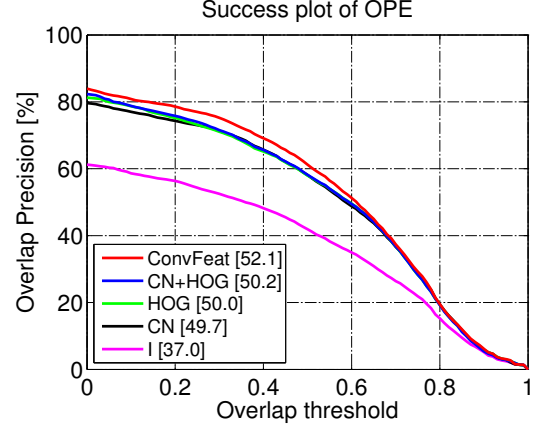

Figure 3. Comparison of the first layer convolutional features with different handcrafted features: HOG, CN and I (image intensity).

the task of accurately locating the target, which is crucial for the tracking problem. Interestingly, the final (fifth) convolutional layer provides a significant performance gain compared to the fourth layer. This is likely due to the high level features encoded by the deepest layers in the network. The final convolutional layer, which has recently been successfully applied in image classification [6], provides a large amount of invariance while still discriminative. In summary, our results suggest that the initial convolutional layer provides the best performance for visual tracking.

We employ the first layer layer for the remainder of our experiments. Figure 3 shows a comparison of the first convolutional layer with hand-crafted features commonly employed in correlation-based trackers. We compare with using grayscale intensity (I), Histogram of Oriented Grandients (HOG) [7] and Color Names (CN) [36]. The success plot displays the mean overlap precision over all 35 color videos in the OTB dataset. Similar results are obtained using HOG and CN. The combination HOG+CN achieves slightly better performance, with an AUC of 50.2%. However, the convolutional features provides improved performance, with an AUC of 52.1%. The activations of the various convolutional features are shown in figure 4.

4.2. State-of-the-art Comparison on OTB

We evaluate the impact of using the convolutional features in the DCF (section 3.1) and SRDCF (section 3.2) approaches. We name our trackers DeepDCF and Deep-SRDCF respectively. For the DeepSRDCF, we reduce the feature dimensionality of the first layer to 40 using Principal Component Analysis (PCA). The PCA basis is computed in the first frame and then remains fix through out the sequence. Our trackers are evaluated on the full OTB dataset (containing 50 videos) and compared with 15 state-of-the-art trackers: SRDCF [10], DSST [9], KCF [21], SAMF [28], ACT [12], TGPR [17], MEEM [40], Struck

Figure 4. Visualization of the employed first-layer convolutional features with the highest energy. Activations are shown for two sample patches (left), taken from the *motorBike* (top row) and *soccer* (bottom row) sequence respectively. The convolutional features capture different colors and edges over image regions.

	ASLA	Struck	KCF	DSST	SAMF	TGPR	MEEM	SRDCF	DeepDCF	DeepSRDCF
OP	56.4	58.8	62.3	67	69.7	62.6	68.7	*78.1*	75.9	**79.4**
DP	59.2	68.7	74.0	74.0	77.7	70.6	79.8	*83.8*	81.8	**84.9**

Table 2. The mean Overlap Precision (OP) and Distance Precision (DP) in percent on the OTB dataset containing all 50 videos. The two best results are shown in red and blue respectively. We only report the results of the top 10 performing trackers.

[20], CFLB [16], MIL [3], CT [41], TLD [23], DFT [34], EDFT [15], ASLA [22].

Table 2 shows the mean Overlap Precision (OP) and Distance Precision (DP) on the OTB dataset. DP is computed as the percentage of frames in a sequence with a center location error smaller than 20 pixels. The DCF based trackers DSST and SAMF, employing HOG and HOG+CN features, provide a mean OP of 67.0% and 69.7% respectively. Our DeepDCF achieves a mean OP of 75.9%, outperforming DSST and SAMF by 8.9% and 6.2% respectively. By using the SRDCF framework, our DeepSRDCF achieves a significant gain of 3.5% mean OP over the DeepDCF. We further improve over the SRDCF by 1.3% in mean OP. Similar conclusions are drawn using Distance Precision (DP). The success plot on the full OTB dataset is shown in figure 5.

4.2.1 Attribute based comparison

We evaluate our tracker by providing an attribute-based analysis on the OTB dataset. The sequences in the dataset are annotated with 11 different attributes: Occlusion, out-of-plane rotation, in-plane rotation, low resolution, scale variation, illumination variation, motion blur, fast motion, background clutter, out-of-view and deformation. Figure 6 shows success plots of four different attributes: scale variation, in-plane rotation, fast motion and occlusion. For clarity, only the top ten trackers in each attribute plot are shown. Both our DeepSRDCF and DeepDCF trackers achieves superior performance compared to the existing methods. In case of scale variation, the standard SRDCF method obtains an AUC score of 59.3%. Our proposed deepSRDCF provides a gain of 4.3% compared to the standard SRDCF
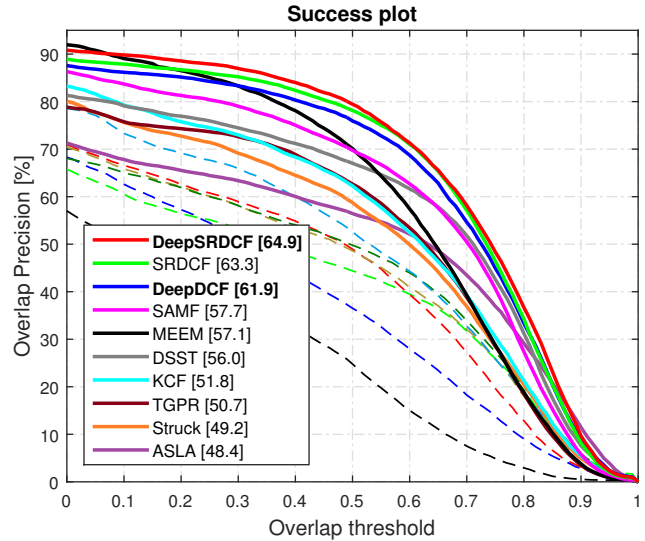

Figure 5. Success plot showing a comparison of our trackers with state-of-the-art methods on the OTB dataset contining all 50 videos. The area-under-the-curve (AUC) scores for the top 10 trackers are reported in the legend.

approach with hand-crafted features. In case of in-plane rotation, the two DCF based trackers SRDCF and DSST provides the best results among existing trackers. Our approach based on deep features and SRDCF achives the best performance with an AUC score of 60.2%. Similarly, our deepSRDCF approach obtains favorable results for fast motion and occlusion, compared to existing trackers.

4.3. State-of-the-art Comparison on VOT2015

The visual object tracking (VOT) challenge is a competition between short-term, model-free visual tracking algorithms. For each sequence in the dataset, a tracker is evaluated by initializing it in the first frame and then restarting the tracker whenever the target is lost (i.e. at a tracking failure). The tracker is then initialized a few frames after the occurred failure. The trackers in VOT are evaluated in terms of an accuracy score and a robustness score. These scores
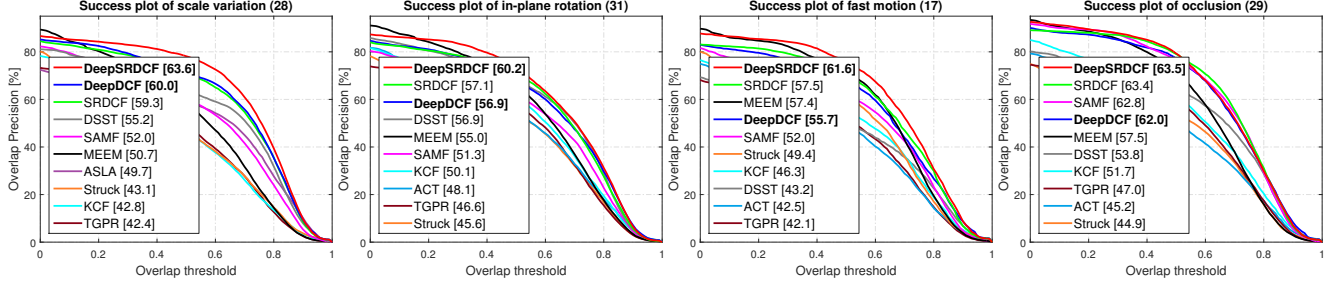
Figure 6. Attribute-based comparison of our trackers with some state of-the-art methods on the OTB-2013 dataset. We show success plots for four attributes: scale variation, in-plane rotation, fast motion and occlusion. The number in each plot title indicates the amount of sequences associated with a particular attribute. Our trackers provide consitent improvements compared to existing methods.
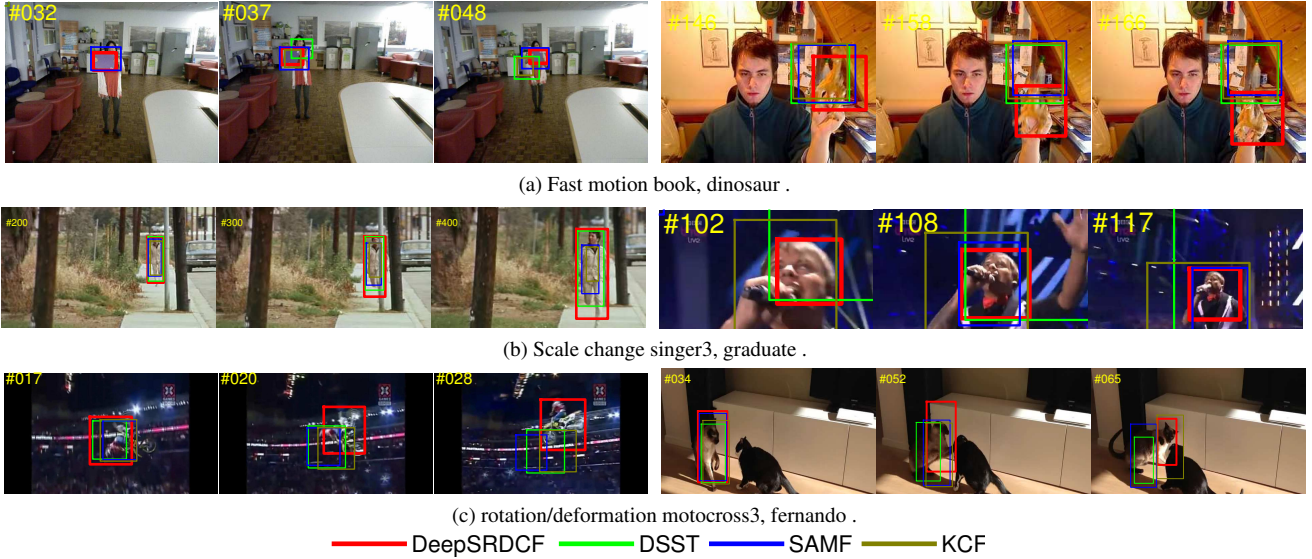

(a) Fast motion book, dinosaur .

(b) Scale change singer3, graduate .

(c) rotation/deformation motocross3, fernando .

DeepSRDCF ——— DSST ——— SAMF ——— KCF

Figure 7. Comparison of our proposed deepSRDCF tracker with the top three trackers of the VOT2014 challenge on example frames from the VOT2015 dataset. The image sequences show challenging situations such as fast motion (top row), scale changes (middle row), rotations and deformations (bottom row).

are computed based on the ground-truth overlap and failure rate measures respectively. The trackers are then ranked in terms of accuracy and robustness for each sequence individually. These ranks are finally averaged to produce the final ranking scores. We refer to [24] for a detailed description of the VOT evaluation methodology.

We provide a comparison of our trackers with 11 state-of-the-art trackers on VOT2015 [1]. In the comparison, we include the top three performing methods of VOT2014 (DSST, SAMF and KCF) and the top 5 existing methods in our OTB comparison (SRDCF, SAMF, MEEM, DSST and KCF). Table 3 shows the results reported by the VOT2015 toolkit [1]. The first two columns contain the mean overlap score and failure rate over the dataset. The remaining columns report the accuracy, robustness and final rank for each tracker. Our DeepSRDCF achives the best final rank on this dataset. Figure 7 shows example frames from the VOT 2015 dataset. Figure 8 shows a visualization of the overall results on the VOT2015 dataset.

4.3.1 State-of-the-art Comparison on ALOV300++

The ALOV300++ dataset includes 314 sequences collected from the internet. Results on this dataset are presented in terms of survival curves, as suggested in [35]. The survival curve of a tracker is constructed by plotting of the F-score value for each video in a descending order. For each video, the F-score is computed based on the percentage of successfully tracked frames, using an intersection-over-union overlap threshold of 0.5. A higher F-score indicates better performance. For more details on the ALOV300 dataset we refer to [35].

We compare our trackers with the 19 methods evaluated in [35]. We additionally include the top 6 existing trackers in our OTB evaluation, namely SRDCF, SAMF, MEEM, DSST, KCF and TGPR. Figure 9 contains the survival curves of all trackers. We also report the average F-score for the top 10 trackers in the legend. Our DeepSRDCF performs favorably compared to the SRDCF with an average F-score of 0.796 compared to 0.787.
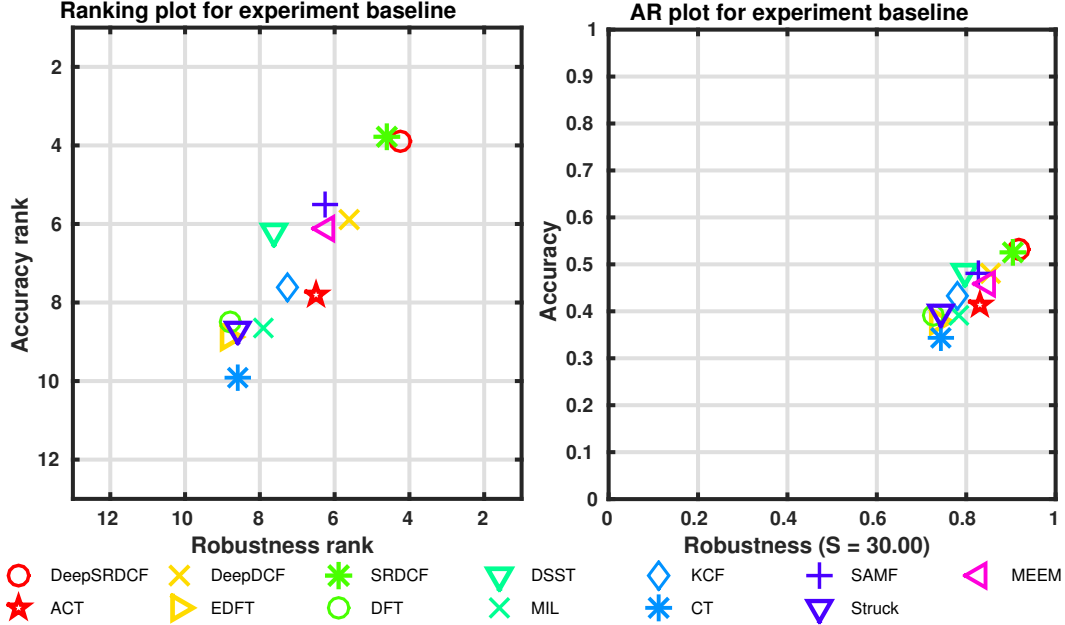
Figure 8. A state-of-the-art comparison on the VOT2015 benchmark. In the ranking plot (left) the accuracy and robustness rank for each tracker is displayed. The AR plot (right) shows the accuracy and robustness scores.

	Overlap	Failure rate	Acc. Rank	Rob. Rank	**Final Rank**
DeepSRDCF	**0.53**	**1.05**	*3.89*	**4.17**	**4.03**
SRDCF	*0.53*	*1.24*	**3.77**	*4.60*	*4.19*
DeepDCF	0.48	*1.75*	5.87	*5.61*	*5.74*
SAMF	0.48	2.05	*5.52*	6.27	5.89
MEEM	0.46	2.05	6.11	6.23	6.17
DSST	*0.48*	2.56	6.20	7.63	6.92
ACT	0.41	2.05	7.81	6.48	7.14
KCF	0.43	2.51	7.60	7.28	7.44
MIL	0.39	3.32	8.67	7.92	8.29
DFT	0.39	4.32	8.50	8.79	8.64
Struck	0.40	3.59	8.70	8.60	8.65
EDFT	0.38	4.08	8.88	8.83	8.85
CT	0.34	4.08	9.91	8.57	9.24

Table 3. The results generated by the VOT2015 benchmark toolkit. The first two columns contains the mean overlap score and failure rate over the entire dataset. The accuracy and robustness ranks are reported in the third and fourth column. The trackers are ordered by their final rank (last column). Our approach provides the best result on this dataset.

5. Conclusions

In this paper, we investigate the impact of convolutional features for visual tracking. Standard DCF based approaches rely on hand-crafted features for robust image description. We propose to use convolutional features within the DCF based framework for visual tracking. We show the impact of convolutional features on two DCF based frameworks: the standard DCF and the recently proposed SRDCF. To validate our proposed tracker, we perform comprehensive experiments on three public benchmarks: OTB, ALOV300++ and VOT 2015. We show that the first convolutional layer provides the best results for tracking, this
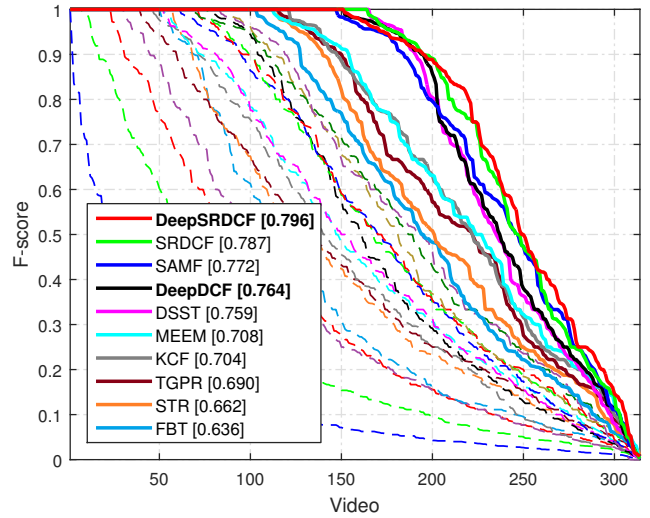

Figure 9. Comparison with state-of-the-art trackers on the ALOV300++ dataset in terms of survival curves. The mean F-scores for the top 10 trackers are provided in the legend. On this dataet, our DeepSRDCF obtains favorable results compared to the standard SRDCF with hand-crafted features.

is suprising considering that the deeper layers are known to be better for general object recognition. We compare our proposed approach with some state of the art methods and obtain state of the art results on three benchmark datasets.

References

[1] The visual object tracking (VOT) challenge 2015. http://www.votchallenge.net.

[2] H. Azizpour, J. Sullivan, and S. Carlsson. Cnn features off-the-shelf: An astounding baseline for recognition. In *CVPRW*, 2014.

[3] B. Babenko, M.-H. Yang, and S. Belongie. Visual tracking with online multiple instance learning. In *CVPR*, 2009.

[4] D. S. Bolme, J. R. Beveridge, B. A. Draper, and Y. M. Lui. Visual object tracking using adaptive correlation filters. In *CVPR*, 2010.

[5] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman. Return of the devil in the details: Delving deep into convolutional nets. In *BMVC*, 2014.

[6] M. Cimpoi, S. Maji, and A. Vedaldi. Deep convolutional filter banks for texture recognition and segmentation. In *CVPR*, 2015.

[7] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005.

[8] M. Danelljan, G. Häger, F. S. Khan, and M. Felsberg. Coloring channel representations for visual tracking. In *SCIA*, 2015.

[9] M. Danelljan, G. Häger, F. Shahbaz Khan, and M. Felsberg. Accurate scale estimation for robust visual tracking. In *BMVC*, 2014.

[10] M. Danelljan, G. Häger, F. Shahbaz Khan, and M. Felsberg. Learning spatially regularized correlation filters for visual tracking. In *ICCV*, 2015.

[11] M. Danelljan, F. S. Khan, M. Felsberg, K. Granström, F. Heintz, P. Rudol, M. Wzorek, J. Kvarnström, and P. Doherty. A low-level active vision framework for collaborative unmanned aircraft systems. In *ECCVW*, 2014.

[12] M. Danelljan, F. Shahbaz Khan, M. Felsberg, and J. van de Weijer. Adaptive color attributes for real-time visual tracking. In *CVPR*, 2014.

[13] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and F.-F. Li. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009.

[14] J. Fan, W. Xu, Y. Wu, and Y. Gong. Human tracking using convolutional neural networks. *IEEE Transactions on Neural Networks*, 21(10):1610–1623, 2010.

[15] M. Felsberg. Enhanced distribution field tracking using channel representations. In *ICCV Workshop*, 2013.

[16] H. K. Galoogahi, T. Sim, and S. Lucey. Correlation filters with limited boundaries. In *CVPR*, 2015.

[17] J. Gao, H. Ling, W. Hu, and J. Xing. Transfer learning based visual tracking with gaussian process regression. In *ECCV*, 2014.

[18] A. Geiger, M. Lauer, C. Wojek, C. Stiller, and R. Urtasun. 3d traffic scene understanding from movable platforms. *PAMI*, 36(5):1012–1025, 2014.

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

[20] S. Hare, A. Saffari, and P. Torr. Struck: Structured output tracking with kernels. In *ICCV*, 2011.

[21] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista. High-speed tracking with kernelized correlation filters. *PAMI*, 2015.

[22] X. Jia, H. Lu, and M.-H. Yang. Visual tracking via adaptive structural local sparse appearance model. In *CVPR*, 2012.

[23] Z. Kalal, J. Matas, and K. Mikolajczyk. P-n learning: Bootstrapping binary classifiers by structural constraints. In *CVPR*, 2010.

[24] M. Kristan, R. Pflugfelder, A. Leonardis, J. Matas, and et al. The visual object tracking VOT 2014 challenge results. In *ECCVW*, 2014.

[25] A. Krizhevsky, I. Sutskever, and G. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012.

[26] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012.

[27] H. Li, Y. Li, and F. Porikli. Deeptrack: Learning discriminative feature representations by convolutional neural networks for visual tracking. In *BMVC*, 2014.

[28] Y. Li and J. Zhu. A scale adaptive kernel correlation filter tracker with feature integration. In *ECCV Workshop*, 2014.

[29] L. Liu, C. Shen, and A. van den Hengel. The treasure beneath convolutional layers: Cross-convolutional-layer pooling for image classification. In *CVPR*, 2015.

[30] K. Nummiaro, E. Koller-Meier, and L. J. V. Gool. An adaptive color-based particle filter. *IVC*, 21(1):99–110, 2003.

[31] M. Oquab, L. Bottou, I. Laptev, and J. Sivic. Learning and transferring mid-level image representations using convolutional neural networks. In *CVPR*, 2014.

[32] S. Oron, A. Bar-Hillel, D. Levi, and S. Avidan. Locally orderless tracking. In *CVPR*, 2012.

[33] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *IJCV*, pages 1–42, April 2015.

[34] L. Sevilla-Lara and E. G. Learned-Miller. Distribution fields for tracking. In *CVPR*, 2012.

[35] A. Smeulders, D. M. Chu, R. Cucchiara, S. Calderara, A. Dehghan, and M. Shah. Visual tracking: An experimental survey. *PAMI*, 36(7):1442–1468, 2014.

[36] J. van de Weijer, C. Schmid, J. J. Verbeek, and D. Larlus. Learning color names for real-world applications. *TIP*, 18(7):1512–1524, 2009.

[37] A. Vedaldi and K. Lenc. Matconvnet – convolutional neural networks for matlab. *CoRR*, abs/1412.4564, 2014.

[38] N. Wang and D. Yeung. Learning a deep compact image representation for visual tracking. In *NIPS*, 2013.

[39] Y. Wu, J. Lim, and M.-H. Yang. Online object tracking: A benchmark. In *CVPR*, 2013.

[40] J. Zhang, S. Ma, and S. Sclaroff. MEEM: robust tracking via multiple experts using entropy minimization. In *ECCV*, 2014.

[41] K. Zhang, L. Zhang, and M. Yang. Real-time compressive tracking. In *ECCV*, 2012.

[42] X. Zhou, L. Xie, P. Zhang, and Y. Zhang. An ensemble of deep neural networks for object tracking. In *ICIP*, 2014.