Hierarchical Convolutional Features Fusion for Visual Tracking

Fan Zhang¹,²*, Shuo Chang³

¹Key Laboratory of the Ministry of Education for Optoelectronic Measurement Technology and Instrument, Beijing Information Science and Technology University, Beijing, 100101, China
²School of Information and Communication Engineering, Beijing Information Science and Technology University, Beijing, 100101, China
³Beijing University of Posts and Telecommunications, Beijing, 100081, China.
*Corresponding author’s e-mail: zhangfan2015@bupt.edu.cn

Abstract. Hierarchical convolutional features have different impact on the tracking performance, as the higher convolutional layers encode the semantic information of targets and earlier convolutional layers are more precise to localize targets. In this paper, we propose a novel scheme for hierarchical convolutional features fusion for visual tracking. In the proposed scheme, hierarchical convolutional features are first concatenated to form the cascading feature at the feature level, and then a convolutional layer is added to reduce the feature dimension. Discriminative correlation filter (DCF) is finally utilized to obtain the target location, which is treated as a differentiable layer in the neural network. The experimental results demonstrate that our proposed scheme achieves superior performances on the visual tracking benchmark.

1. Introduction

Within the field of computer vision, visual tracking is a fundamental problem and has attracted more and more attentions from various applications, such as automated surveillance and vehicle navigation [1][2]. Without priori knowledge of the tracking target, the discriminative information needs to be learned online to achieve high performance. Despite huge progress in recent years, visual tracking is challenging due to significant appearance changes, cluttered-background, as well as illumination changes, etc.

Due to the availability of large-scale visual dataset and computation power improvement, convolutional neural networks (CNNs) have achieved superior performance on a wide range of computer vision tasks, e.g. image classification and object detection [3]. Inspired by the strong representation power of CNNs [4], some tracking algorithms based on CNNs [5][6] have been developed recently. However, the amount of annotated datum to train CNNs is scarce because only the target bounding box in the first frame is available in visual tracking. Hence, previous works utilized the offline CNN pretrained on ImageNet [7], e.g. VGG-Net [8], to obtain the feature representation of the target directly. In this way, the dilemma of insufficient training samples is avoided. Meanwhile, the powerful representation ability of convolutional features is exploited.

Hierarchical convolutional features have different impact on the tracking performance. As shown in Figure 1, the last layers of CNNs are more effective to capture semantics, but they are too coarse to precisely localize targets due to less fine-grained spatial details. In addition, the earlier layers are
precise in localization, while they are less invariance to appearance changes. This property suggests that fusing multi-layer features of CNNs is of great significance in visual tracking. This is motivated by the fact that semantics of higher layers are robust to appearance variations, and features of earlier layers are precious in localization to alleviate drifting.

For this purpose, Ma et al. proposed to use features from hierarchical layers of CNNs to encode the target appearance (CF2) [6]. Specially, CF2 learned correlation filters on each convolutional layer and inferred the target position with a coarse-to-fine searching approach. In essence, linear weighting of each layer correlation response, i.e. decision level fusion, was adopted.

In spite of strong representation capacity of hierarchical convolutional features, the required computational resources can not be ignored. Specially, the higher the dimension of convolutional features, the higher the computational complexity. Therefore, it is necessary to reduce the feature dimension. However, CF2 utilizes all the feature maps from multiple layers with high computational complexity.

Based on above analyses, we want to fuse hierarchical convolutional features with low feature dimension and high performance, and our contribution focuses on the feature dimension reduction strategy for high-dimensional convolutional features with lightweight network.

In this paper we study how to fuse hierarchical convolutional features and reduce the feature dimension after fusion. A novel scheme named as hierarchical convolutional features fusion (HCF) for visual tracking is proposed. Specifically, hierarchical convolutional features are first concatenated at feature level to form the cascading feature with a high feature dimension, and then a convolutional layer with size 1 × 1 is added to realize the feature dimension reduction. DCF is finally utilized to obtain the target location, which is treated as a differentiable layer in the neural network and the back propagation is derived. The experimental results show that our scheme obtains superior performances on the benchmark.

The rest of the paper is organized as follows. First, Section 2 reviews the formulation of DCF. Our proposed work is presented in Section 3. The experimental results are given in Section 4, and finally conclusions are drawn in Section 5.

2. Discriminative Correlation Filter

Discriminative correlation filter (DCF) based trackers have achieved superior performances in terms of accuracy and real-time performance. Concretely, DCF regresses all the shifted versions of input to a Gaussian function, and the target position is predicted by searching the maximum value of correlation response. Furthermore, fast Fourier transform (FFT) is utilized to realize computational efficiency based on the circular correlation operation, and limited training samples are fully utilized due to the circularity. In the formulation of DCF, a discriminative regression is trained based on the target patch feature $\phi(x)$ and desired response $y$. Specifically, the filter $w$ is obtained by minimizing the following cost function:

$$
\varepsilon(w) = \| \sum_{l=1}^{D} w^l \ast \phi^l(x) - y \|_2^2 + \lambda \sum_{l=1}^{D} \| w^l \|_2^2
$$

(1)

where $w^l$ is the $l$-th channel of filter $w$, $\phi^l(x)$ is the $l$-th channel of $\phi(x)$, $\ast$ denotes circular convolution, $y$ is set to a Gaussian function with the peak placed at the target center location, and the constant $\lambda \geq 0$ is the regularization coefficient. The solution is obtained as follows:

$$
\hat{w}^l = \frac{(\phi^l(x))^\ast \circ \widehat{\phi}}{\sum_{l=1}^{D} (\phi^l(x) \circ (\phi^l(x))^\ast + \lambda)}
$$

(2)
where the hat denotes the 2-dimensional discrete Fourier transform (DFT), and * represents the complex conjugate operator, and ⊙ denotes the Hadamard product.

At the detection stage, the search patch features \( \varphi(z) \) centered at the previous target location in the new frame are extracted, and the detection scores at all locations are computed as follows:

\[
s = \mathcal{F}^{-1}(\sum_{i=1}^{D} \hat{\varphi}^i(\tilde{x}))^*
\]

(3)

where \( z \) denotes the search patch extracted from the current image region and \( \mathcal{F}^{-1} \) is the inverse DFT (IDFT). At last, the target location can be obtained with the highest detection score \( z \).

Note that, the computational complexity of DCF is related to the feature dimension \( D \). The higher the feature dimension, the higher the computational complexity. Therefore, it is necessary to reduce the feature dimension for the purpose of reducing complexity. In this paper, a \( 1 \times 1 \) convolutional layer is added to reduce the number of channels for convolutional features.

3. Hierarchical Convolutional Features Fusion for Visual Tracking

3.1. The framework of HCF

The objective of our work is to fuse hierarchical convolutional features to take advantage of the spatial details of earlier layers and semantics of higher layers, while reducing the feature dimension. Therefore, a novel framework is proposed to realize this purpose, as shown in Figure 2. Specifically, the hierarchical convolutional features of \( t \)-th frame are extracted from a pre-trained CNN model, and then multi-layer convolutional features are concatenated to form the cascading feature. Next, a convolutional layer with size \( 1 \times 1 \) is added to realize the purpose of reducing feature dimension. Discriminative correlation filter (DCF) is finally utilized to obtain the target location, which is treated as a differentiable layer in the neural network. The details of the back propagation derivation are described in Section 3.3.

3.2. Convolutional Features

In this work, we use the convolutional features extracted from CNN pre-trained on ImageNet, e.g. VGG16, to encode the target appearance. With the increase of the depth of convolutional layers, the spatial resolution is reduced due to the pooling operations in the CNNs. For instance, the outputs of the conv3-4, conv4-4, and conv5-4 of VGG16 are 56 × 56, 28 × 28, and 14 × 14, respectively. The hierarchical convolutional features are concatenated to exploit the superiority of earlier and higher layers simultaneously. However, the spatial resolution is different considering hierarchical convolutional features. We resolve the issue by resizing the feature maps to a fixed size with bilinear interpolation.

3.3. DCF Derivation

The Figure 3 is the histogram of the activation values of conv5-4 for all the feature maps within the object region. From Figure 3, we can conclude that most of the feature maps have zero or small values.
within the object region. Therefore, the convolutional features are sparse and it is necessary to reduce the feature dimension.

![Feature maps histogram of conv5-3.](image)

Figure 3. Feature maps histogram of conv5-3.

For this purpose, we propose to reduce the feature dimension with a convolutional layer with size of $1 \times 1$. Our aim is to learn the parameters of the convolutional layer in an end-to-end fashion. Given the features of the first frame $\varphi(x, \theta)$, the desired response $y$ should get a high response at the target location. So the objective function is formulated as:

$$L(\theta) = \| g - y \|^2 + \gamma \| \theta \|^2$$

(4)

where

$$g = F^{-1}(\sum_{l=1}^{L} g^{l}(\mathcal{Q}(\hat{\varphi}^{l}(x, \theta))^*))$$

(5)

$$\hat{g}^{l} = \frac{\delta F^{l} (\sum_{k=1}^{K} \mathcal{Q}(\hat{\varphi}^{l}(x, \theta))^*)}{\delta \theta}$$

(6)

Now, let’s derive the backward formulas. According to [9], the gradient of DFT and IDFT are formulated as follows:

$$\hat{g} = F(g)$$

(7)

$$\frac{\delta F^{-1}(g^{l})}{\delta \varphi} = F^{-1}(\frac{\delta F(g^{l})}{\delta \varphi})$$

(8)

$$\frac{\delta F^{-1}(g^{l})}{\delta \bar{\varphi}} = F^{-1}(\frac{\delta F(\bar{g}^{l})}{\delta \bar{\varphi}})$$

(9)

Using above expression, the per-element derivative of the forward pass is calculated as:

$$\frac{\delta L}{\delta \varphi^{uv}(x)} = (F(\frac{\delta L}{\delta \varphi^{uv}}))^{uv}$$

(10)

For the back propagation process, $\hat{\varphi}^{l}_{uv}(x)$ and $(\hat{\varphi}^{l}_{uv}(x))^*$ are treated as independent variable as follows:

$$\frac{\delta L}{\delta \varphi^{l}_{uv}(x)} = \frac{\delta L}{\delta \hat{\varphi}^{l}_{uv}(x)} = \frac{\delta L}{\delta \hat{\varphi}^{l}_{uv}} \frac{\delta \hat{\varphi}^{l}_{uv}(x)}{\delta \varphi^{l}_{uv}(x)} (\hat{\varphi}^{l}_{uv}(x))^*$$

(11)

$$\frac{\delta L}{\delta \hat{\varphi}^{l}_{uv}(x)} = \frac{\delta L}{\delta \hat{\varphi}^{l}_{uv}} \frac{\delta \hat{\varphi}^{l}_{uv}(x)}{\delta \hat{\varphi}^{l}_{uv}(x)} (\hat{\varphi}^{l}_{uv}(x))^* = \frac{\delta L}{\delta \hat{\varphi}^{l}_{uv}} \frac{\delta \hat{\varphi}^{l}_{uv}(x)}{\delta \hat{\varphi}^{l}_{uv}(x)} (\hat{\varphi}^{l}_{uv}(x))^* + \gamma \frac{\delta L}{\delta \hat{\varphi}^{l}_{uv}(x)}$$

(12)

Therefore, the derivation of $\varphi^{l}(x)$ is reformulated as:

$$\frac{\delta L}{\delta \varphi^{l}(x)} = \frac{\delta L}{\delta \hat{\varphi}^{l}(x)} + \gamma \frac{\delta L}{\delta \hat{\varphi}^{l}(x)}$$

(13)

Once the error is propagated backwards to the convolutional layer, the rest of the back propagation can be done as traditional CNN optimization. Since the training sample in the first frame is scarce, we utilize data augmentation to alleviate this problem. Data augmentation is a standard strategy which can
improve the generalization of the model. Typically, the following data augmentation techniques are considered:

- **Rotation**: Rotation from a fixed set of 19 angels ranging from $-45^\circ$ to $45^\circ$.
- **Blur**: Blur with a Gaussian filter with standard deviation of 0.5, 0.9, 1 and 1.2, which simulates the motion blur and scale variations.

### 3.4. Model Update Strategy

At the training stage, the model is updated by updating the numerator $A^t$ and denominator $B^t$ of the correlation filter $\hat{w}^t$ with a learning rate $\gamma$, respectively:

\[
\hat{w}^t = \frac{A^t}{B^t + \lambda}
\]

\[
A^t = (1 - \gamma)A^t_{t-1} + \gamma((\hat{x}^t(x))^\ast \hat{y})
\]

\[
B^t = (1 - \gamma)B^t_{t-1} + \gamma \sum_{k=1}^{\lambda} \hat{x}^t(x) \hat{x}^t(x)^\ast
\]

where $t$ is the frame index.

### 4. Experiments

We validate our proposed method by conducting comprehensive experiments on OTB-2015 [10]. The success plot is used to evaluate all the trackers, which illustrates the percentage of frames where the intersection-over-union overlap with the ground truth exceeds a threshold. The tracking schemes are ranked using the area under curve (AUC) displayed in the legend.

#### 4.1. Implementation Details

Our tracker is implemented in Matlab and the implementation details are presented as follows. VGG-Net-19 [8] network is employed to extract convolutional features, which uses the MatConvNet library [11]. The fully-connected layers are removed, and the outputs of conv3-4, conv4-4, and conv5-4 are utilized as our features. The extracted features are multiplied by a Hann window to mitigate boundary discontinuities. The regularization parameter is $\lambda = 10^{-2}$, and the kernel width is 0.1 for generating the Gaussian function labels. The learning rate is set to 0.01. The dimensionality of cascading feature is reduced to 640 using our proposed approach. The convolutional layer parameters are computed in the first frame, and fixed throughout the sequence.

#### 4.2. OTB-2015 Dataset

We compare the proposed algorithm with three trackers, which serve the same purpose of hierarchical convolutional features fusion: CON (the hierarchical convolutional features are concatenated without feature dimension reduction), CON-PCA (the hierarchical convolutional features are concatenated using principal component analysis (PCA) as dimensional reduction strategy), and CF2 [6].

The comparison on the OTB-2015 is provided, and Figure 4 shows the success plot under one-pass evaluation (OPE) over all the 100 videos. From the success plot in Figure 4, our proposed HCF obtains an AUC score of 58.6%, which outperforms the CF2 by 1.9%. In the case of none-feature dimension reduction (all the 1280 dimensional features are used), the AUC score decrease by 4.4%. This indicates that not all the convolutional features are related to the tracking task, and the noisy feature maps may degrade the tracking performance. The CON-PCA method obtains an AUC score of 54.5%, which decrease by 4.1% compared with HCF. In conclusion, HCF is more effective than unsupervised PCA, and CF2 which fuses hierarchical convolutional features with linear weighting of each layer correlation response. Namely, feature level fusion is superior than decision level fusion.
Figure 4. Success plot showing a comparison with three trackers on OTB-2015.

**Attribute-based evaluation** The tracking performances under different attributes are further analyzed on OTB-2015, and the success plots of six different attributes are in Figure 5.

From Figure 5, the following observations are concluded. First, our method outperforms the compared trackers on 11 attributes. This benefit comes from the proposed end-to-end feature dimension reduction strategy and data augmentation technique. Second, our method provides a gain of 2.1%, 1.5%, 1.5%, 2.6%, 3% and 5% compared to CF2 in the case of in-plane-rotation, out-of-plane rotation, motion blur, scale variation, occlusion, and low resolution, respectively. This is attributed to considering the rotation and blur in data augmentation.
Figure 5. Attribute-based evaluation of our algorithm on OTB-2015 dataset. Success plots are shown for four attributes using OPE. Each plot title consists of the number of videos and the respective attribute.

### 4.3 Qualitative Evaluation

Some tracking results of HCF and CF2 on two challenging video sequences are presented. The qualitative evaluation results are demonstrated in Figure 6. The video sequences from top to down are: Box and Kitesurf. From Figure 6, we can conclude that the proposed algorithm achieves superior performance compared to CF2. Since the correlation among hierarchical convolutional features is not exploited, the CF2 fails in sequence when occlusion and out-of-plane rotation occur (Box and Kitesurf). However, our tracker performs well in most of the scenarios.

Figure 6. Qualitative comparison of HCF and CF2 on four challenging sequences of OTB-2015. (from top to down are Box and Kitesurf, respectively.)

### 5. Conclusion

In this paper, we have proposed a scheme named as hierarchical convolutional features fusion for visual tracking. Our proposed scheme takes advantage of semantics and spatial details simultaneously, and provides a general feature dimension reduction technique for DCF-based trackers. The experimental results show that our scheme has achieved superior performance. As the future work, we plan to introduce attention mechanism to learn more effective features.
Acknowledgments
This work was supported in part by the Fundamental Research Funds in Beijing Information Science and Technology University (No. 2025019), National Natural Science Foundation of China (No.61702099), and Project: the Research of Multi-antenna Reader Distributed Positioning System (No. 2020KYNH213).

References
[1] Liu, S., Liu, X., Wang, S., et al. (2020) Fuzzy-aided solution for out-of-view challenge in visual tracking under IoT-assisted complex environment. Neural Computing and Applications, 16: 1-9.
[2] Du, F., Liu, P., Zhao, W., et al. (2019) Joint Channel Reliability and Correlation Filters Learning for Visual Tracking. IEEE Transactions on Circuits and Systems for Video Technology, 99: 1-10.
[3] Zhao, X., Li, W., Zhang, Y., et al. (2017) A faster rcssn-based pedestrian detection system. In: Vehicular Technology Conference. Toronto. pp. 1–5.
[4] Alex, K., Ilya, S., and Geoffrey, E. H. (2012) Imagenet classification with deep convolutional neural networks. In: Neural Information Processing Systems. Lake Tahoe. pp. 1097–1105.
[5] Jack, V., Luca, B., Henriques, J., et al. (2017) End-to-end representation learning for correlation filter based tracking. In Computer Vision and Pattern Recognition. Hawaii. pp. 5000–5008.
[6] Ma, C., Huang. J. B., Yang, X., et al. (2016) Hierarchical convolutional features for visual tracking. In: IEEE International Conference on Computer Vision. Las Vegas. pp. 3074–3082.
[7] Jia, D., Wei, D., Richard, S., et al. (2009) Imagenet: A large-scale hierarchical image database. In: Computer Vision and Pattern Recognition. Miami. pp. 248–255.
[8] Karen, S., Andrew, Z. (2015) Very deep convolutional networks for large-scale image recognition. In: International Conference on Learning Representations. Diego. pp. 1–14.
[9] Christoph, B., Patrick, H., Lukas, D., et al. (2017) On the computation of complex-valued gradients with application to statistically optimum beamforming. arXiv preprint, arXiv:1701.00392.
[10] Yi, W., Jongwoo, L. Object tracking benchmark. (2015) IEEE Transactions on Pattern Analysis and Machine Intelligence, 37: 1834–1848.
[11] Andrea, V., Karel, L. (2015) Matconvnet: Convolutional neural networks for matlab. In: the 23rd ACM International Conference on Multimedia. California. pp. 689–692.