Using Domain-Knowledge for Correlation-based Face Tracking

Marko Kastelic
Fakulteta za elektrotehniko
Univerza v Ljubljani
mk5534@student.uni-lj.si

Abstract

Visual object tracking is an important problem in computer vision. It is becoming even more important with the introduction of autonomous vehicles. The basic idea is to track the target with only the initial position given. The first breakthrough was made with the introduction of correlation filter-based tracking. The performance of tracking improved both in terms of accuracy and computational efficiency. One recent breakthrough was seen with the proposal of Convolutional Neural Networks (CNN) to extract features from the target patch. Every year results are improving on different challenges related to object tracking. However, because of the fact that there are many different targets, no specific domain knowledge can be used. In this paper we tackle the case with a single target to see if we can improve results by using domain knowledge of faces. Specifically, we investigate if using a CNN-based feature extractor originally trained for a face classification ensures better performance in face tracking than using a general purpose CNN.

We are testing the trackers on eight sequences from the Visual Tracker Benchmark (VTB) dataset, where the target is face. Obtained results suggest that the tracker with the knowledge of faces can match the performance of the tracker with the general purpose CNN but does not improve it.

1. Introduction

Object tracking is an important part of computer vision. The basic goal is to track a target in a video through time with only the initial state given. The target is tracked with an unsupervised algorithm, which predicts the target position in the next frame. Video surveillance is one of the most important applications of object tracking, but it is also used in many other fields such as robotics, autonomous driving and entertainment. In the recent years, several of the proposed algorithms have made a significant progress in this field. Despite that, there are still problems that have not been solved completely. The main obstacles for successful long-term tracking are illumination variation, change of target's appearance (*e.g.* pose, size, blur caused by a fast motion) and occlusion. These obstacles cause predicted location to slowly drift from the target's actual location and if the drift becomes too large, the location of the target is completely lost - the intersection of predicted and actual location of the target becomes zero. Re-initialization of the tracking algorithm remains an open problem in object tracking. When evaluating the performance of trackers, robustness is one of the often-used metrics. It measures the quality of tracking through time and is defined as the number of times the position of the target is completely lost and the tracker needs to be re-initialized manually (failure rate). Other common metrics is accuracy, which measures quality of prediction in each frame. It is defined as the overlap between the predicted region and the actual region of a target.

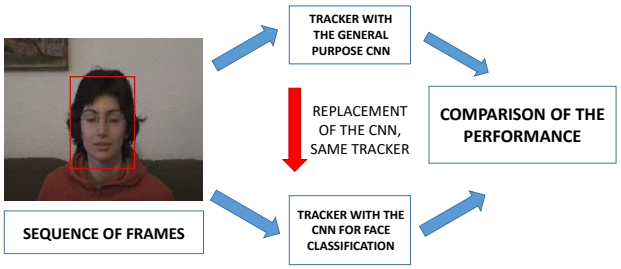


Figure 1: The basic idea of this work: we are using one of the trackers from the VOT Challenge for the task of face tracking. We want to see if the performance can improve when using neural networks with domain knowledge of faces.

Another important aspect of a tracker is its computational complexity. A big step in this area was made with the proposition of correlation filter-based tracking [1]. Correlation by definition measures similarity between 2 signals,

images, *etc*. Filters are applied to image or features with convolution. In the time domain computation of convolution is complex. However, one of the important properties is the convolution theorem, which states that convolution in the frequency domain can be computed as an element-wise multiplication. The main idea of the correlation filter-based tracking is to find filter that produces high response in target region and low in the background. The peak in the response map can then be used for prediction of the new location. It is important to update the filter through time, so that tracking can adjust to changes of the target.

A correlation filter can be applied directly on a raw image, however applying it on features extracted from the image makes tracking more robust. These features can be various descriptors such as Histogram of Oriented Gradients (HOG), SIFT, Color Names *etc*. In the recent years, features obtained from convolutional neural networks are replacing hand-crafted descriptors. Trackers using CNNs are achieving state-of-the-art results on object tracking challenges like VOT Challenge [1].

On these challenges trackers are usually evaluated on a large number of sequences that contain different targets, *e.g*. vehicles, people, faces, balls. Thus, extracted features must be general for the tracker to perform well on all the targets. In this paper we instead focus on tracking a single type of target - a human face. We want to see what happens with the performance of tracking when using domain knowledge of faces. We are using tracker from the VOT Challenge and replacing the general neural network used in their tracker, which was trained to classify various types of objects, with a network designed for face classification to see if the performance of face tracking can be improved. The basic idea of this work is illustrated in figure 1.

Both versions of the tracker are evaluated on eight sequences from the VTB dataset. The main contributions of this work are:

- we are testing trackers on one specific target to see if domain knowledge of the target can improve the performance
- we show that results on sequences with face as the target can be matched using the knowledge of faces but are not improved compared to the best performing trackers

2. Related Work

This section is split into two parts. In the first part we give an overview of the methods using correlation filters including the recent CNN-based approaches. Since we are focusing on tracking of faces we mention some of the works related to deep neural networks for face classification in the second part.

2.1. Video Tracking with Correlation Filters

Bolme *et al*. [1] were the first to propose the use of correlation filters for video tracking. Due to the use of the filter in the frequency domain, convolution becomes an element-wise multiplication, which reduces computational costs. Consequently, this approach is very fast and appropriate for the use in real-time. By using MOSSE filter, their approach was also robust to appearance changes of the target. This was improved further by Li *et al*. [12] who proposed a multi-view correlation filter-based tracker that reduced the influence of both target and background changes. This was done with the use of multi-view features to select a stable view of the target. They also present a mechanism to detect scale variations. To make tracking more robust to scale changes, Danelljan *et al*. [5] proposed scale-pyramids to train classifier, which are used to estimate the scale independently of the target translation. Li and Zhu [13] proposed a scale adaptive kernel for the correlation filter. They also integrated additional features such as HOG and ColorNames to further improve the performance. To deal with occlusion, part-based tracking strategies [14, 15] were proposed. Henriques *et al*. [10] proposed circulant matrices, which are used to create negative samples by translating the base (positive) sample. These samples can then be used for learning the correlation filter using discriminative methods. The underlying assumption is that the samples are periodic, which enables the use of the Fast Fourier Transform (FFT). Using the FFT makes these methods fast, however violation of the periodic assumption creates unwanted boundary effects [7]. This leads to some limitations of these methods, such as a restricted search region. Danelljan *et al*. [7] proposed a spatially-regularized discriminative correlation filter, which addresses above-mentioned problems. Galoogahi *et al*. [9] proposed a background-aware correlation filter with real negative examples being used rather than the shifted ones. With the use of the information from the background they also addressed the problems with the background clutter and occlusions. Ma *et al*. [17] proposed the use of the target and its context to form a template for long-term tracking.

In recent years, convolutional neural networks replaced hand-crafted features like HOG and color intensity [16, 6]. Choi *et al*. [3] proposed an attentional correlation filter network, which uses multiple correlation filters and adaptively selects only the most relevant ones. Valmadre, Bertinetto *et al*. [23] proposed integration of the correlation filter into the neural network, making it one of its layers. This way parameters of both CNN and correlation filter can co-adapt, enabling end-to-end learning of parameters. Their results have not made significant improvement in the per-

[1] http://votchallenge.net/

formance compared to the state-of-the-art methods. However, their neural network was smaller and had fewer parameters, which makes their method appropriate for embedded systems, where the amount of memory is smaller. Ma *et al*. [16] proposed using outputs from multiple CNN layers, rather than just the last one, to adaptively learn correlation filters. This way, results of the predicted location are more accurate. Nam and Ham [18] proposed multi-domain CNN.

2.2. Face Recognition with Deep Neural Networks

As in most computer vision problems in recent years, deep neural networks have also been used for the task of face recognition and classification. Parkhi *et al*. [19] showed that deep CNNs with appropriate training can achieve competitive results. Sun *et al*. [21] used deep networks to learn a face representation. Zhang and Zhang from Microsoft Research [25] proposed multi-task CNN, where they trained face and non-face decision together with pose estimation and facial landmark localization. Sun *et al*. [22] have constructed a DeepID2+ network with an improved performance and robustness to the image corruption. Researchers from Google [20] used 200 million face identities [19] to train CNN and achieved 99.63% accuracy on the Labeled Faces in the Wild dataset.

3. Methodology

For the purpose of analysis, whether using the domain knowledge of a specific target (in our case face) can improve the performance of tracking, we are testing Efficient Convolution Operators for Tracking (ECO) tracker [4] from VOT Challenge 2017. It is a modified version of the Convolution Operators (C-COT) tracker [8] and is based on a continuous correlation filter-based tracking. They are using multiple features for description of the target: HOG descriptors, Color Names and features extracted from a CNN. They are using ImageNet-VGG neural network [2]. This network was designed for the classification of 1,000 different objects and is well suited for different challenges, such as VOT, where the tracker is tested on various types of objects. Since we are only interested in tracking faces, we are replacing the ImageNet-VGG with VGG-Face neural network [19], which is primarily used for face classification, to see if it improves the performance of the tracker.

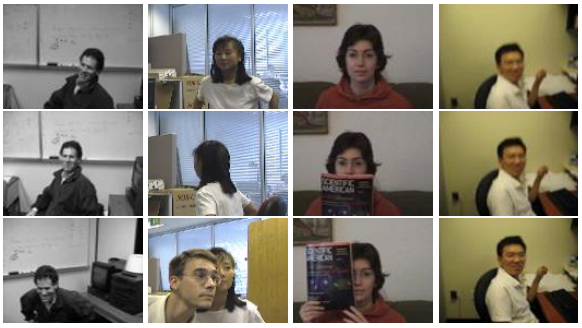
3.1. Correlation Filter-based Tracking

The basic idea of correlation filter-based tracking is using a correlation filter on an input to get a response map. The peak in this map is used as the prediction for the new position of the target. The correlation filter can be applied directly on the raw image, but usually due to the different changes *e.g*. illuminance, pose of a target, motion blur, some features are extracted from the image first. These features can be different descriptors such as HOG and Color Names, or features obtained from CNNs. Applying the filter on the features instead of raw images makes tracking more robust. Then correlation filter is used on these features via convolution. One of the important properties of convolution is Convolution Theorem:

$$\mathcal{F}\{f * g\} = \mathcal{F}\{f\} \cdot \overline{\mathcal{F}\{g\}}, \tag{1}$$

where $\mathcal{F}\{\}$ denotes Forier Transform, $*$ stands for convolution, and $\overline{\mathcal{F}}$ for complex conjugate of Fourier Transform. According to this theorem, in a 2D discrete space convolution in the frequency domain is calculated as an element wise multiplication - dot product ($\odot$). This leads to significantly smaller computational complexity. Fourier transform can also be computed efficiently with the use of the Fast Fourier Transform algorithm.

Before applying the correlation filter on images or features, filter has to be trained. At each time step, features are extracted from the input image. Convolution between the filter and features is computed next to obtain the response map. The peak in this map is used as the prediction for the new location. Features are extracted in the new location. These features are used in combination with desired correlation output to calculate and update the correlation filter. The first breakthrough in the use of correlation filters was made by Bolme *et al*. [1] who proposed the Minimum Output Sum of Squared Error (MOSSE) training method. The basic idea of this method is to compute a filter *h* that has the minimal sum of squared errors between the actual and desired corre-



(a) FleetFace (b) Girl (c) FaceOcc1 (d) BlurFace

Figure 2: Sample frames from the Visual Tracker Benchmark dataset. Each column has 3 samples from one sequence. In the first column are frames from the *FleetFace* sequence, face is well visible. The second column contains samples from the *Girl* sequence, face is occluded with another face, there is only the back of the head visible. In the third column are frames from the *FaceOcc1* sequence, a large part of the face is occluded. In the last column are representative frames from *BlurFace* sequence, position of face is changing quickly due to camera motion (3 samples shown are consecutive).

Parameter	conv1	conv2	conv3	conv4	conv5	fc6	fc7	fc8
filter size	7×7×3×96	5×5×96×256	3×3×256×512	3×3×512×512	3×3×512×512	6×6×512×4096	1×1×4096×2048	1×1×2048×1000
stride	2	2	1	1	2	1	1	1
pad	0	1	1	1	0, 1	0	0	1

Table 1: Overview of the 8-layer ImageNet-VGG architecture. Filter size represents dimensions of a filter - width, height, depth and number of filters; sliding of the filter is defined by the stride (number of pixels) and pad defines zero-padding around the border of the input volume.

Parameter	conv1-1	conv1-2	conv2-1	conv2-2	conv3-1	conv3-2	conv3-3	conv4-1
filter size	3×3×3×64	3×3×64×64	3×3×64×128	3×3×128×128	3×3×128×256	3×3×256×256	3×3×256×256	3×3×256×512
stride	1	1	1	1	1	1	1	1
pad	1	1	1	1	1	1	1	1
parameter	conv4-2	conv4-3	conv5-1	conv5-2	conv5-3	fc6	fc7	fc8
filter size	3×3×512×512	3×3×512×512	3×3×512×512	3×3×512×512	3×3×512×512	7×7×512×4096	1×1×4096×4096	1×1×4096×2622
stride	1	1	1	1	1	1	1	1
pad	1	1	1	1	1	0	0	0

Table 2: Overview of the 16-layer VGG-Face architecture. Filter size represents dimensions of a filter - width, height, depth and number of filters; sliding of the filter is defined by the stride (number of pixels) and pad defines zero-padding around the border of the input volume.

lation output. As explained before, because of the smaller computational complexity, calculation of the optimal filter is done in the frequency domain. Let big letters denote the Fourier Transform ($F = \mathcal{F}\{f\}$ - input, $H = \mathcal{F}\{h\}$ - filter, $G = \mathcal{F}\{g\}$ - desired output). The optimization problem in mathematical terms is:

$$\min_{\overline{H}} = \sum_i ||F_i \odot \overline{H} - G_i||^2, \tag{2}$$

whrere *i* is the index of the training image. The solution to this problem is:

$$\overline{H} = \frac{\sum_i G_i \odot \overline{F_i}}{\sum_i F_i \odot \overline{F_i}}. \tag{3}$$

The desired or ground truth correlation response is defined by a 2D Gaussian distribution. Another similar method is Average of Synthetic Exact Filters. Here one filter is computed at each time step and the final filter is the average of all filters. Some other proposed filters are Kernelized Correlation Filters and Dense Spatio-Temporal Context filter.

After training (initializing) the filter, it can be applied on a test sequence. However, due to the different changes of the target and background it is important to update the filter online so that it can adapt to all this changes. In the MOSSE method, correlation filter is updated with every new frame and exponential forgetting principle is applied. On the other hand, in the ECO tracker we are using, the updates are less frequent. This way they get better results, which they attribute to a lower degree of over-fitting. Beside that it can also make the tracker more robust to rapid changes.

3.2. ImageNet-VGG Neural Network

ImageNet-VGG network can be used for object classification. It is trained on 1,000 categories from the ImageNet dataset. It consists of five convolutional and three fully-connected layers. The size of an input image has to be 224 × 224 pixels. There are three different architectures: fast (CNN-F), medium (CNN-M), and slow (CNN-S). In the ECO tracker, a modified medium version is used. The architecture is presented in the table 1. Normalization is performed with the subtraction of an average image obtained as the average pixel values from images in the training set.

3.3. VGG-Face Neural Network

VGG-Face network is designed for classification of 2,622 individuals. Its output is a score for each of the 2,622 classes. It consist of five convolutional blocks, which have multiple convolutional layers, and three fully-connected layers. The size of the input image is 224 × 224 pixels. The average image is given as one average value for each of 3 color channels. Filter sizes and other parameters are given in the table 2.

4. Experiments

In this section we present the data relevant to our experiments. In the first subsection composition of the dataset is given, next experimental protocol is described and in the last part of this section metrics used for the evaluation are presented.

4.1. Dataset

In this paper we are testing trackers on the Visual Tracker Benchmark dataset [24], which was designed with the purpose of objective evaluation of the trackers. The dataset contains 100 sequences from different sources and consists of different targets *e.g.*, people, faces, cars. In several of those sequences the target is a face and we are using some of them in this paper. All of the sequences we are using have at least a few hundred frames. Some examples from the dataset are presented in figure 2. The following list contains names of the sequences we are using and a few basic characteristics of each sequence:

- **FleetFace**: 707 frames with 720x480 resolution, face is well visible in all the frames
- **Girl**: 500 frames, 128x96 resolution, girl's head is rotating through the frames, thus on times, the face is completely invisible (back of the head), on some frames there is also an occlusion with another face
- **BlurFace**: 493 frames, 640x480 resolution, position of the faces changes rapidly due to the camera motion
- **FaceOcc1**: 892 frames, 352x288 resolution, a large part of the face is occluded with the magazine
- **Trellis**: 569 frames, 320x240 resolution, person is moving, but watching into the camera in almost all frames
- **Boy**: 602 frames, 640x480 resolution, person is jumping across the hall
- **David2**: 537 frames, 320x240 resolution, person is slowly moving
- **Jumping**: 313 frames, 352x288 resolution, person is jumping on the spot

4.2. Experimental protocol

The tracker and neural networks used in this paper have been already pre-trained, thus all of the data can be used for evaluation of the trackers. The data consist of eight sequences as described in section 4.1. No pre-processing of the data was done since trackers are searching for a target in the whole frame.

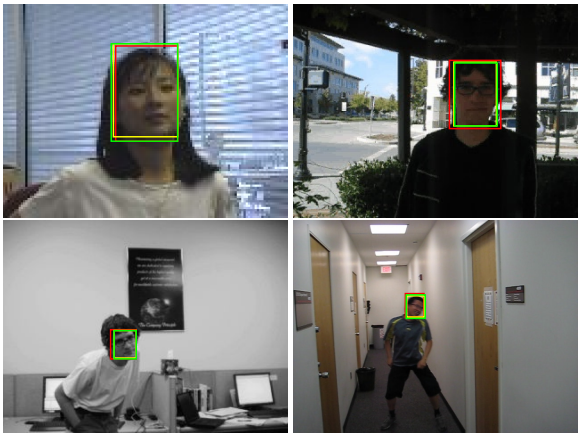


Figure 3: Demonstration of results. Red belongs to the ground truth, yellow to the tracker with ImageNet-VGG and green to the tracker with VGG-Face network. Results are close to the ground truth and generally include more or less the whole face.

Both versions of the tracker are tested on all sequences and the overall performance of the trackers is estimated. A special importance is given to the comparison of performance. The goal is to find out if there is any improvement when using the tracker with the CNN designed for face classification.

4.3. Performance metrics

Since we are using the tracker from the VOT Challenge, we are also using well-defined and established performance metrics used in VOT. Their code for performance evaluation was proposed by Kristan *et al*. [11] and is publicly available.

In this paper [11] they propose two methods for evaluation: accuracy and robustness. The accuracy evaluates the quality of performance based on the predicted and ground truth bounding boxes; it measures the overlap between both bounding boxes. Accuracy at time t is calculated with the following equation:

$$\phi(t) = \frac{A^T(t) \cap A^G(t)}{A^T(t) \cup A^G(t)} \tag{4}$$

where $A^G(t)$ is the ground truth bounding box and $A^T(t)$ is the bounding box predicted by the tracker. $\cap$ stands for intersection and $\cup$ for union. The average accuracy can be computed in all frames and another variant is expected accuracy, which is defined as the average accuracy in some validation frames:

$$\phi = \frac{1}{N_{val}} \sum_{j=1}^{N_{val}} \phi(j) \tag{5}$$

On the other hand, robustness measures the quality of the tracker through time and is defined as the number of

times prediction completely drifts from the target and thus requires tracker's re-initialization. This happens when there is no overlap between predicted and ground truth bounding boxes. To estimate robustness, the tracker must be evaluated in a supervised mode, so that it can be re-initialized. Kristan *et al*. [11] proposed that re-initialization does not happen immediately after the failure, but after some number of frames. This way the probability that the tracker will fail again in the next few frames is smaller.

	Sequence							
Network	BlurFace	Boy	David2	FaceOcc1	FleetFace	Girl	Juping	Trellis
ImageNet-Vgg	0.57	0.65	0.78	0.76	0.62	0.77	0.50	0.72
Vgg-Face	0.57	0.65	0.70	0.77	0.62	0.77	0.47	0.76
	Average (expected overlap)							
ImageNet-Vgg	0.768							
Vgg-Face	0.766							

Table 3: Performance in terms of tracking. The average accuracy is given for both trackers for every sequence and the overall performance is reported as the expected overlap. We can see that there are no significant differences in the performance.

Network	ImageNet-Vgg	Vgg-Face
Speed [fps]	0.9726	0.2304

Table 4: Comparison of the speed

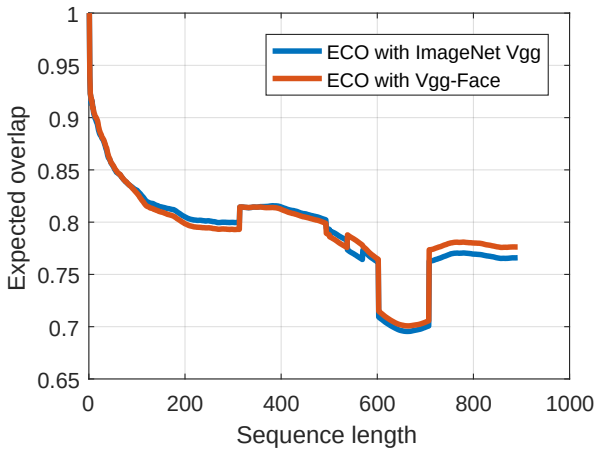


Figure 4: Expected overlap on long-term. The expected overlap is more or less the same in both versions of the tracker; in the longer sequence, the overlap for the second tracker (VGG-Face network) is slightly higher, but not significant.

5. Experimental Results

5.1. Implementation details

As described before, the ECO tracker combines different features extracted from a target patch. They are using a fast variant of the HOG algorithm with cell size $n_c = 10$ and the standard 9 bins for orientation. For Color Names they are using cell size $n_c = 4$. The filter is updated in every sixth frame. For features extracted using the CNN, we first tested the tracker using one of the first convolutional layers as the first output and a higher-lying fully-connected layer as the second output. The idea was to get some general features from the first output and some high-level, more explicit features of a face from the second output. The receptive field is becoming bigger with higher layers, and thus features are more specific. However, we found out that the performance is better using the second output from the convolutional layers rather than one of the fully-connected layers. The reasons for this may be that features from the last layers are too explicit for our problem, because there are some disturbances such as occlusion. In this case, less specific features may be more appropriate. This was the case using both the ImageNet-VGG and VGG-Face network. For the final evaluation we are using features from *conv1* (normalized output) and *conv5* (relu output) layers as the output, as was the original setup in the ECO tracker. And for VGG-Face network, we are using layers *conv1-2* and *conv5-2* as the output layers.

The evaluation was performed in Matlab using single 2-core CPU (Intel Core i7-7500U 2.90 Ghz)

5.2. Results

We used the toolkit implementation from the VOT Challenge for the evaluation. We tested trackers in supervised and unsupervised mode to get robustness and accuracy for both versions of the ECO tracker. Results are somewhat surprising as they show there is no major difference in the performance of both trackers. In fact, original tracker with the ImageNet-VGG network even had a few per mills better performance in terms of the average accuracy. Some examples are shown in figure 3. There are some minor differences in accuracy for separate sequences, but overall performance is more or less the same. Results of both trackers for all sequences are given in table 3

In our opinion these results are the consequence of several factors. First, the target (face) is relatively large in all frames. Even though there are some disturbances like occlusion or fast motion, generally speaking, faces are still relatively well visible. This results in a good performance of both trackers for long and short-term tracking. Both trackers never drift from the target and do not need to be re-initialized. As a result they achieve highest score for robustness (tracking never failed). This also means that the performance is equal in supervised and unsupervised test and that accuracy remains relatively high through time as can be seen in figure 4. On different challenges like VOT, targets

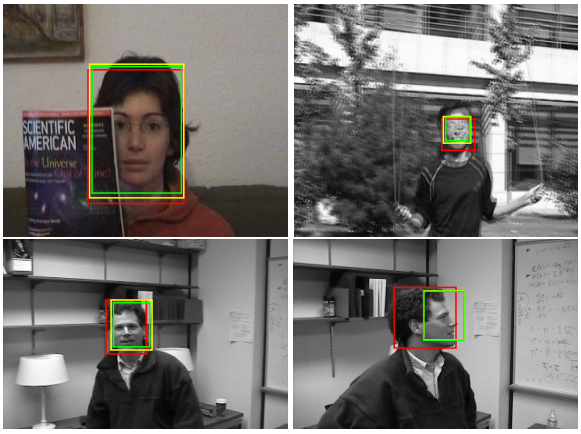


Figure 5: Examples showing ground truth regions and predicted regions. Red is ground truth, yellow and green belong to the tracker using the with ImageNet-VGG and VGG-Face network respectively. Examples show that results with the baseline comparison are dependent on the ground truth. Both trackers predicted the face correctly but the predicted region is smaller than that of the ground truth and resulting in a lower accuracy. The last image demonstrates this especially well. Examples also show that there are some minor differences between both trackers, regions predicted by the tracker using the knowledge of faces (green rectangle) are often slightly smaller.

are of different sizes, some of them very small, and there is a bigger probability that the tracker will fail. Another important thing is the rigidity of a face, which means that the scale remains more or less the same through time. The scale would only be changing if the target would be moving away or coming closer relative to the camera, which is not the case in our sequences. As a result, there is no problem with the scale estimation.

Another important aspect is the choice of the tracker. The ECO tracker is one of the best performing trackers, which means it is harder to make improvements. They also use a combination of features for the description of the target, which means that the difference in terms of performance is harder to see when using a different CNN. They are using a modified and improved version of the correlation filter-based tracking and the performance of their tracker is very good on our test sequences. The predicted region is more or less inside the ground truth region and covers the face. This means that the score (accuracy) is dependent on how the labeling of the data is performed. With a different set of the ground truth labels accuracy might be even better. Since the predicted region more or less covers the face, it is hard to improve results. In our sequences the ground truth region is sometimes a little bigger than the face and maybe on some other data, that would cover only the face, results

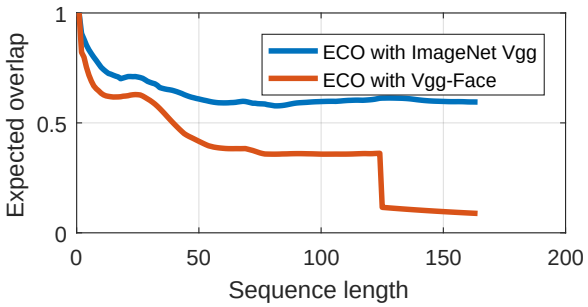


Figure 6: Performance on the non-face sequence. Results are expected and show worse performance of the tracker with the knowledge of faces.

with the VGG-Face network might be slightly better compared to ImageNet-VGG. A few examples are in figure 5, where we can see that the predicted region is smaller than the ground truth region for both trackers. Region predicted by the VGG-Face network is also a little smaller than the region predicted by the tracker that uses ImageNet-VGG. To see if there are any bigger changes in the performance when using one or the other network, we would have to test a tracker that uses only the CNN features.

Another important aspect of the performance is the tracker's speed. Since the VGG-Face network has more layers, this has a negative effect on the speed. A relatively large downfall in the speed was observed comparing the two networks. Speed comparison in terms of frames per second is given in the table 4.

To see if the choice of the network has any influence on the results we tested both trackers on additional non-face sequence (the Motocross sequence from the VOT Challenge). Results are unsurprising and show significantly worse performance of the tracker with domain knowledge of faces. Expected overlap in long-term is presented in figure 6. From this we can conclude that tracker with the knowledge of faces can match the results of the general purpose tracker when the target is face and have worse results on the non-face targets.

6. Conclusion

In this paper we investigated if the domain knowledge of faces can improve the performance of face tracking. We used the ECO tracker from the VOT Challenge and replaced their ImageNet-VGG neural network with VGG-Face network designed for face classification. Obtained results did not show any improvement. One of the reasons for this could be that the original ECO tracker itself already achieves good results. We used features from two layers of the VGG-Face network, it is possible that the performance would be improved if we would add a third output layer.

References

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

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

[3] J. Choi, H. J. Chang, S. Yun, and T. Fischer. Attentional correlation filter network for adaptive visual tracking. In *CVPR*, 2017.

[4] M. Danelljan, G. Bhat, F. S. Khan, and M. Felsberg. Eco: Efficient convolution operators for tracking. In *CVPR*, 2017.

[5] M. Danelljan, G. Hager, F. S. Khan, and M. Felsberg. Accurate scale estimation for robust visual tracking. In *BMVC*, 2014.

[6] M. Danelljan, G. Hager, F. S. Khan, and M. Felsberg. Convolutional features for correlation filter based visual tracking. In *ICCV*, 2015.

[7] M. Danelljan, G. Hager, F. S. Khan, and M. Felsberg. Learning spatially regularized correlation filters for visual tracking. In *ICCV*, 2015.

[8] M. Danelljan, A. Robinson, F. S. Khan, and M. Felsberg. Beyond correlation filters: Learning continuous convolution operators for visual tracking. In *ECCV*, 2016.

[9] H. K. Galoogahi, A. Fagg, and S. Lucey. Learning background-aware correlation filters for visual tracking. In *ICCV*, 2017.

[10] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista. High-speed tracking with kernelized correlation filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(3):583–596, 2015.

[11] M. Kristan, J. Matas, A. Leonardis, T. Vojir, R. Pflungfelder, G. Fernandez, G. Nebehay, F. Porikli, and L. ehovin. A novel performance evaluation methodology for single-target trackers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38, 2016.

[12] X. Li, Q. Liu, Z. He, H. Wang, C. Zhang, and W. S. Chen. A multi-view model for visual tracking via correlation filters. *Knowledge-Based Systems*, 113:88–99, 2016.

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

[14] S. Liu, T. Zhang, X. Cao, and C. Xu. Structural correlation filter for robust visual tracking. In *CVPR*, 2016.

[15] T. Liu, G. Wang, and Q. Yang. Real-time part-based visual tracking via adaptive correlation filters. In *CVPR*, 2015.

[16] C. Ma, J. B. Huang, X. Yang, and M. H. Yang. Hierarchical convolutional features for visual tracking. In *ICCV*, 2017.

[17] C. Ma, X. Yang, C. Zhang, and M. H. Yang. Long-term correlation tracking. In *CVPR*, 2015.

[18] H. Nam and B. Han. Learning multi-domain convolutional neural networks for visual tracking. In *CVPR*, 2016.

[19] O. M. Parkhi, A. Veldadi, and A. Zisserman. Deep face recognition. In *BMVC*, 2015.

[20] F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering. In *CVPR*, 2015.

[21] Y. Sun, X. Wang, and X. Tang. Deep learning face representation from predicting 10,000 classes. In *CVPR*, 2014.

[22] Y. Sun, X. Wang, and X. Tang. Deeply learned face representations are sparse, selective, and robust. In *CVPR*, 2015.

[23] J. Valmadre, L. Bertinetto, J. Henriques, A. Vedaldi, and P. H. S. Torr. End-to-end representation learning for correlation filter based tracking. In *CVPR*, 2017.

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

[25] C. Zhang and Z. Zhang. Improving multiview face detection with multi-task deep convolutional neural networks. In *WACV*, 2014.