LEARNING A LIGHTWEIGHT CONVOLUTIONAL NEURAL NETWORK FOR VISUAL TRACKING AND FACIAL ATTRIBUTE ANALYSIS

ZHU LINNAN

M.Phil

The Hong Kong

Polytechnic University

2017

The Hong Kong Polytechnic University

Department of Computing

Learning A Lightweight Convolutional Neural Network for Visual Tracking and Facial Attribute Analysis

ZHU LINNAN

A thesis submitted in partial fulfillment of the requirements for the degree of Master of Philosophy

July 2016

Certificate of Originality

I hereby declare that this thesis is my own work and that, to the best of my knowledge and belief, it reproduces no material previously published or written, nor material that has been accepted for the award of any other degree or diploma, except where due acknowledgement has been made in the text.

_____ (Signature)

ZHU LINNAN (Name of Student)

Abstract

In this thesis, we study the problems of object tracking and facial attribute analysis, in particular age and gender recognition. For object tracking, recently CNN based trackers have been proposed to improve tracking performance. Despite achieving state-of-the-art performance, existing CNN trackers still have many drawbacks. 1) Most of these methods utilize two separated CNNs for each input, while this strategy will increase much the number of model parameters, which consequently requires more labeled samples at the training stage. 2) Some CNN trackers can run at over 100 fps on GPU, but run very slowly on CPU due to the high complexity of network structure. In order to deal with these issues, in this thesis we propose a novel frame-pair based CNN architecture, which can balance tracking speed and accuracy. Instead of adopting two-stream CNNs, we fuse frame pairs in the input stage, resulting in a single-stream CNN tracker with much fewer parameters. The proposed tracker can learn generic motion patterns of objects with less video data compared with previous CNN based methods. The evaluation is conducted on the VOT14, OTB50 and OTB100 benchmark datasets. The proposed tracker achieves competitive results with state-of-the-arts but with much less memory and complexity. Our tracker can track objects in a speed of over 100 (30) fps with a GPU (CPU), much faster than most existing CNN based trackers.

For age and gender recognition, CNN based methods have achieved state-of-the-art accuracy but they are time consuming for mobiles or low-end PCs for the following two issues. 1) Complex CNN architecture. Most of CNN based methods directly employ the popular architectures (e.g., AlexNet and VGG), which are very complex and overdesigned for

age and gender recognition. 2) Regarding age and gender recognition as two independent problems. Actually, age and gender recognition are two highly correlated tasks about facial attributes, and it will be beneficial if we can optimize these two tasks together. In this thesis, we propose a lightweight deep model to recognize age and gender from a face image via a joint regression model. Specifically, our model employs a multi-task learning scheme to learn shared features for these two correlated tasks in an end-to-end manner. Extensive experimental results on the recent Adience benchmark demonstrate that our model achieves competitive recognition accuracy with the state-of-the-art methods but with much faster speed, i.e., about 10 times faster in the testing phase.

Publications

Conference Papers

1. **Linnan Zhu**, Lingxiao Yang, David Zhang and Lei Zhang, “Learning Real-time Generic Tracker Using Convolutional Neural Networks”, International Conference on Multimedia and Expo (ICME), 2017 (**Oral presentation**, selected as Top 3% paper for Finalist of the World’s FIRST 10K Best Paper Award).

2. **Linnan Zhu**, Keze Wang, Liang Lin and Lei Zhang, “Learning a Lightweight Deep Convolutional Network for Joint Age and Gender Recognition”, International Conference on Pattern Recognition (ICPR), 2016. (**Oral presentation**)

Acknowledgement

It is my great honor to be a student of my supervisor Prof. Lei Zhang, who is always there whenever I face any difficulties, no matter in study or life. His patience and continuous support helps to complete this thesis. I appreciate his guide and directions, and am very lucky to be one of his students.

Besides, I would like to thank Prof. Xiangchu Feng, Prof. Wangmeng Zuo, Prof. Cheng Deng, Prof. PengFei Zhu, Prof. Kaihua Zhang, and my seniors, Mr. Keze Wang, Mr. Lingxiao Yang, Mr. Shuhang Gu, Mr. Jun Xu, Mr. Sijia Cai, who are pursuing their doctoral degree in department of Computing, the Hong Kong Polytechnic University. They are all very experienced researchers and love what they are doing and do what they love at the same time. I really appreciate their help and patience when I met with some difficulties. I hope their dreams can be true in the future.

Finally, I really want to say thank you to my family and the friends who have helped me before. Whenever I meet with difficulties, they are always supporting me to overcome the challenges. Because of their support, I can do what I love and only focus on the research study without any other worries.

Contents

List of Figures

List of Tables

List of Major Abbreviations

CNN	Convolutional Neural Network
STC	Spatio-Temporal Context
DSTC	Depth Spatio-Temporal Context
HoG	Histogram of Oriented Gradient
KCF	Kernelized Correlation Filter
DCF	Dual Correlation Filter

Chapter 1

Introduction

1.1 Research Background and Motivation

In this thesis, we mainly focus on two tasks in computer vision applications: object tracking, and facial attribute analysis, especially for age and gender recognition. Real time visual tracking is a core problem in computer vision field, which can be used for surveillance, transportation management (Figure 1.1), self-driving car (Figure 1.2) and etc.

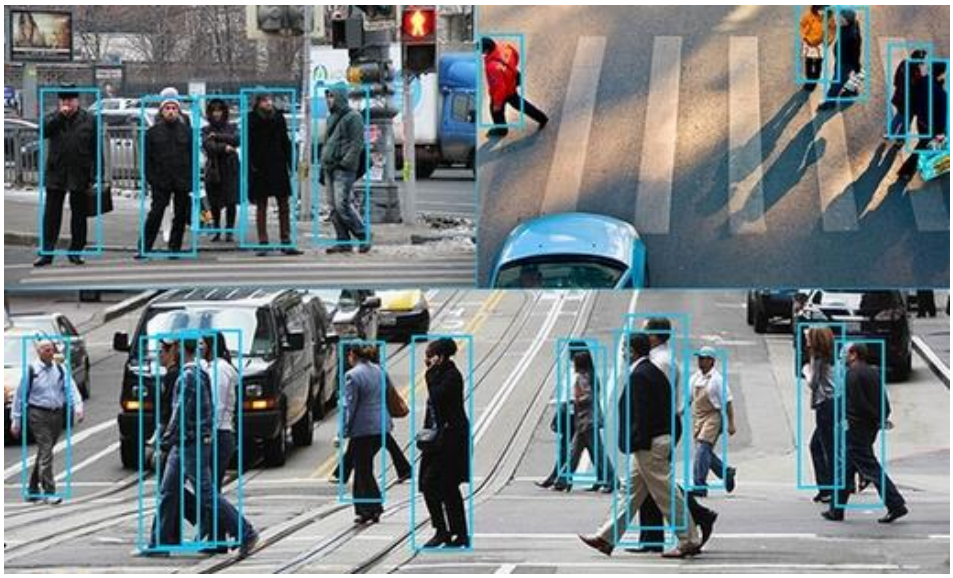

Figure 1.1 Visual tracking application for transportation management

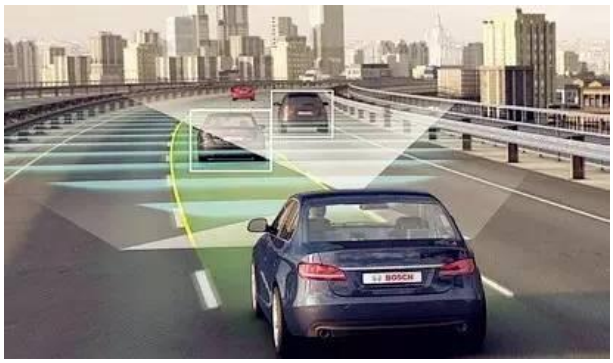

Figure 1.2 Visual tracking application for self-driving car

While visual tracking is still a challenge task due to uncertain changes of objects online, such as illumination changes, shape deformation, occlusion, fast motion and etc. Therefore, it is more difficult to achieve real time visual tracking with an acceptable accuracy. In this thesis, we aim to propose some novel methods to improve the tracking accuracy along with faster speed than the state-of-the-art algorithms.

Real time facial attribute analysis is also a very promising topic. By recognizing some facial attributes (age, gender, race and etc) in a single image can be applied to help companies to build an accurate user profile (Figure 1.3).

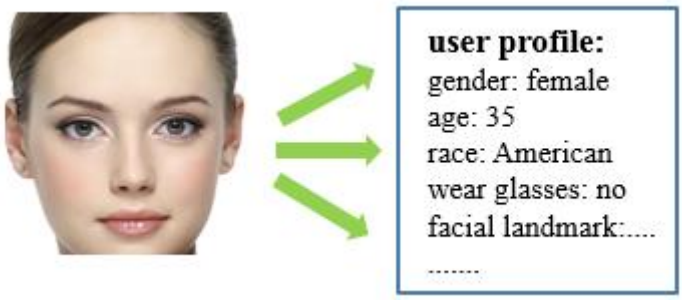

Figure 1.3 Facial attribute analysis application for building user profile

For the case of protecting user privacy, the algorithm is better to be implemented in an embed system, e.g. mobile phone. However, current hand-craft features based methods are not accurate enough for industry level usage, while CNN based methods achieves better accuracy but are very time consuming due to their complicated network structures and regarding two highly correlated tasks as two independent problems. In this thesis, we aim to propose a lightweight deep CNN model to jointly recognize age and gender attributes from a single image.

Object Tracking: Object tracking is always an interesting and active topic in research community. Firstly, researchers put forward tracking-by-detection methods [6], [7], [8], [9], [10], [11], but they are too complex to achieve real-time performance. Later on, Zhang et al. [12], [13] proposed a binary compressed Bayes classifier method and can achieve real-time tracking. Then Zhang et al. [14] applied spatial, temporal and context information, which improves the tracking performance but still cannot handle the partial occlusion and full occlusion problems. Henriques et al. [15] proposed a Kernelized Correlation Filter (KCF), which uses Histogram of oriented gradient (HoG) feature for real-time tracking. However, these real-time tracking methods. Recently, CNN based trackers [23], [24], [25], [26], [27], [28], [29], [30] were proposed to improve the tracking accuracy. Despite achieving state-of-the-art performance, existing CNN trackers still have many drawbacks. Current CNN trackers can run at over 100 fps on a GPU device, but still execute at very slow speed on a single CPU processer (around 5 fps in our PC) due to the complexity of their network structures. Therefore, all CNN trackers cannot be easily integrated into small devices like mobile phones. In order to deal with above issues, in this thesis, we aim to learn a network in

an off-line fashion and track objects online. To make the tracker lighter and incorporate temporal cue, we propose a novel frame-pair based CNN architecture, which can get the balance of tracking speed and accuracy.

Age and Gender Recognition: Real time facial attribute recognition is a very promising and hot topic, especially, recognizing age and gender in a single image has sparked off a great interest in both research community and industrial companies. Recently the popular deep learning technique has been successfully applied to age and gender recognition. Levi et al. [52] proposed to individually train two models for each problem. However, these CNN based methods are time consuming for mobiles or low-end PCs for the following two issues:

1) Exploiting the complex CNN architecture. Most of CNN based methods directly employ the popular architectures (say AlexNet [51] and VGG [53]), which are specially designed for large scale visual recognition, e.g. ImageNet Challenge [54]. The network architectures are very complex and overdesigned for the age and gender recognition, which heavily increases the computation burden.

2) Regarding age and gender recognition as two independent problems. Although in [52] the model they trained for age and gender recognition has a same architecture, the parameters are different and the method requires a complex cascade architecture of deep model. As the matter of fact, age and gender recognition are two highly correlated tasks about facial attributes. It will be beneficial to recognize accuracy and time efficiency if we can optimize these two tasks together.

In order to address above mentioned issues, in this thesis, we propose a lightweight deep

framework to jointly recognize age and gender in a fast end-to-end manner. The proposed framework employs a multi-task learning scheme to complete these two correlated tasks.

1.2 Review of Object Tracking

1.2.1 Review of Traditional Methods for Object Tracking

As is known to all, object tracking is always a quite interesting and active topic in research community. But many influencing factors, such as diverse poses, different intensity of illumination, complicated backgrounds lead object tracking to be a very tough challenge, which attracts a large amount of researchers. In the old times, the approaches proposed by researchers almost used visible light cameras to take photos. Some methods put forward local features, such as gradient features, i.e. HoG feature [3]. And some approaches captured the picture's interest points, such as SIFT feature [4] . Even though these RGB based methods yielded quite good result in object tracking, they are not able to handle some complicated problems, such as the image containing complex background and so on. That is to say, when dealing with these challenges, RGB based approaches can either obtain inaccurate results or greatly add the computational expenses. Some researchers put forward tracking-by-detection methods [6], [7], [8], [9], [10], [11] based on RGB images. The approach in [6], [7] applied a tracker and a detector to deal with the tracking problems. But the detector should be pre-trained, thus it cannot deal with other datasets or real-time environment. Method in [8] contained a large amount of training dataset and approaches in [9], [10], [11] applied very complex procedures to obtain the training datasets, which added the computational cost and resulted in the tracking speed not very fast and cannot extend to real-time tracking. More

importantly, the training and testing phases of these three approaches are absolutely divided so that if one pose is not inside the training datasets, the methods cannot track it in the testing phase.

Later on, Zhang et al [12], [13] proposed a binary compressed Bayes classifier framework for the purpose of solving tracking problems and obtained quite robust results. More importantly, due to using compressive concept, the speed is faster than pervious methods, which can achieve real-time tracking. However, because this approach does not have occlusion handling function, it still has its limitations and cannot tackle heavy occlusion or full occlusion situations. Especially, if the target disappears and then re-appears, this approach will lose the target and cannot capture the target again. Zhang et al. [14] applied spatial, temporal and context information for visual tracking. They proposed a formulation based on Bayesian theory and low level features to quantify the relationship between the context and the location of target center. The framework is simple and very fast, but still has its limitations and cannot handle the partial occlusion and full occlusion problems. Henriques et al. [15], [16] proposed a Kernelized Correlation Filter (KCF) and Dual Correlation Filter (DCF), which uses HoG feature and FFT for fast tracking.

1.2.2 Review of Convolutional Neural Networks for Object Tracking

Recently deep learning techniques are very active. Deep architecture for deep learning is applied to a large amount of issues including image reconstruction, image classification, image de-noising and object recognition etc. Nowadays, CNNs have demonstrated their expressive representation power in high-level recognition problems [17], [18], [19], [20]. In

this section, we briefly discuss existing CNN-based trackers which are close to ours. Many authors attempted to utilize CNNs to address tracking task. Li et al. [21] proposed a CNN architecture to learn the feature representation built upon multiple image cues. Guan et al. [22] tracked objects with their proposed online CNN. The CNNs utilized in [21], [22] were trained from scratch and updated online while they were suffered from a lack of labeled training data. To address this problem, in [21], the authors sampled useful training data from the historical tracked results. In [22], they utilized K-means to learn weights in CNN because K-means does not require any labeled data. Instead of training CNNs from scratch online, our tracker is built upon the pre-trained CNNs and trained in an off-line way.

Many works have exploited powerful pre-trained CNN features for visual tracking [23], [24], [25]. Danelljan et al. [23] demonstrated that with the Kernelized Correlation Filters (KCF) [15], the shallow convolutional layers of the pre-trained CNNs are more appropriate for tracking since they preserve more spatial information compared with deeper layers. Ma et al. [24] independently trained KCF trackers on multiple layers of the CNNs. Then the target location is predicted by combining the decision of all trackers, while the combination weights were hand-crafted. Qi et al. [25] presented an adaptive weighted method that pools KCF trackers from different CNN layers. Unlike above CNN-KCF structure, several works directly trained networks for visual tracking [26], [27]. Wang et al. [26] tracked objects with two networks, called GNet and SNet. They attached GNet and SNet on top of the conv5-3 and conv4-3 layers of the VGG [85] model respectively, and demonstrated that these two networks were able to capture complementary information for visual tracking. Nam and Han [27] proposed a multi-domain CNN (MDNet) and achieved state-of-the-art results on several

benchmarks. In this paper, however, we focus on the problem of learning a generic tracker in an off-line way. When applying our tracker online, it can track objects at a high speed, which is significant faster than all trackers mentioned above. Recently, several authors proposed to learn a similarity function for visual tracking [28], [29], [30]. Tao et al. [28] and Bertinetto et al. [30] considered the tracking as a template matching task and utilized a siamese architecture to learn the similarity function. Held [29] treated the similarity learning as a regression problem and directly regressed the location of tracked objects. There are many differences between our method and aforementioned. While [28], [29], [30] utilized a two-stream framework for visual tracking, we just trained our tracker using a single CNN stream. When using similar CNN architecture, our tracker has less model parameters, which significant reduces the size of training samples, and improves the speed for online tracking.

1.3 Review of Age and Gender Recognition

1.3.1 Review of Age Recognition

For the age recognition task, Kwon et al. [59] firstly published papers in this filed. They applied cranio-facial development theory to distinguish baby and adults by calculating six ratios of distance on frontal face images. Then, Ramanathan et al. [60] attempted eight ratios of distance. Nevertheless, due to the anthropometry concept based models are very sensitive to head pose and do not use the additional texture information, this model is not suitable to distinguish the adults. [61] proposed an active appearance model (AAM) to represent the face image. Lanitis et al. [62] tried different classifiers for age estimation based on their age image representation, especially the quadratic aging function [63]. The advantage of the AAM

model over the previous anthropometry based model is that the AMM model used shape and texture information together and can classify different age groups, not only the young and the adults. Then Geng et al. [64] proposed an AGing pattErn Subspace (AGES) model to learn a personalized pattern for a individual person. However, this method needs a large number of cross-age images of one person. [65] proposed to learn a low dimensional pattern from face images at each age, which makes it much easier and more flexible to build a face database. Afterwards, some feature extraction related algorithms were introduced to this area. The local features, such as texture and shape features [66], and Local Binary Patterns (LBP) [47] were used first, but due to the obvious local variations in facial appearance, some global feature were applied afterwards. Spatially Flexible Patch (SFP) feature put forward by Yan et al. [68] includes the local patches and position information, aiming to help to deal with occlusion, and head pose variations. In the previous study, researchers calculated ratios between different facial features [68]. After that, [64], [69] presented subspace and manifold learning respectively. They obtained good performance on some certain constrained datasets, i.e., the face images are frontal and well-aligned, which are not suitable for the unconstrained face benchmark. The above mentioned approaches are all evaluated on some certain benchmarks, which have a lot of constrains, such as face images need to be captured in a near-front view and with a suitable lighting, well-aligned or other perfect conditions. In a word, these related methods cannot be directly implemented for the real-world challenge.

1.3.2 Review of Gender Recognition

Along the development of age recognition, the research on gender recognition is back to

1990s. Cottrell et al. [70] first proposed a neural network model, but the faces were under constrained. Then Brunelli et al. [71] proposed a HyperBF network for this task by extracting the geometrical features. In [72], Wiskott et al. implemented Gabor wavelets to a face representation. Afterwards, [73] applied principal component analysis (PCA) and linear discriminant analysis (LDA) based on Gabor wavelet to recognize the gender label. Then Sun et al. [74] demonstrated that genetic algorithms (GA) was suitable for gender recognition task and verified the feature selection was at a very important stage for this task. And In [75], they obtained good recognition accuracy in a constrained dataset FERET. This method extracts feature via independent component analysis and uses LDA to be the classifier. Afterwards, Costen et al. [76] proposed a sparse SVM approach and yielded good results in a Japanese face dataset. In [77] researchers implemented SVM and Adaboot classifiers on the raw images, respectively. In the recent years, Webers Local texture Descriptor [50] was introduced in gender recognition task and yielded good performance on a constrained face images dataset.

1.3.3 Review of Convolutional Neural Networks for Age and Gender Recognition

By directly extracting features from raw images, deep CNN models have made impressive progresses on visual recognition problems, e.g., image classification, object detection, semantic segmentation and many other recognition [51], [53]. Inspired by the success of CNNs on visual recognition problems, Levi et al. [52] proposed a deep CNN model to tackle the age and gender recognition problem and obtained significant results by sufficient training

data. Because of the high resolution of input image (227×227) and large convolution filters, the parameters of this network architecture is in a large quantity. Besides, the CNN based approach are complex enough and hardly to implement on a low-end PC or a mobile phone. In order to deal with this problem, in this paper, we propose a lightweight deep model for the age and gender recognition tasks.

1.4 Thesis contributions

Object Tracking:

1) In this thesis, we firstly bring the depth weight function into the tracking problem. This depth weight function sets more weight to the not occluded sections and less weight to the occluded sections. As a result, the proposed depth based algorithm can improve the efficiency and accuracy of tracking especially the occlusion problem. To conclude, I concede the proposed depth spatio-temporal context (DSTC) model achieves the optimal performance than the other approaches [12], [14].

2) Firstly we take place raw pixel feature with HoG feature in STC [14] framework. This step is regarded as coarse tracking. Secondly, we bring the Convolution Neural Network (CNN) to learn a filter from the input HoG feature pair to the output. The second step is treated as fine-grained tracking. Finally, we fuse the confidence map obtained by the above two steps to get the final prediction center point of the target. From the comparison results of Visual Tracking Benchmark Dataset [31], we can see that the proposed

STC-HoG-CNN model achieves the best performance than the STC and DCF approaches. To conclude, I concede the proposed STC-HoG-CNN model achieves the optimal performance than the other approaches [14], [15], [16].

3) In this research, we aim to learn a network in an off-line fashion and track objects online. To make the tracker lighter and incorporate temporal cue, we propose a novel frame-pair based CNN architecture. Specifically, we stack the cropped regions from two successive frames together as the input to a CNN stream. The fusion at early stage allows the tracker directly learn many temporal features [36]. The output of our tracker is a probability map which indicates the location of target. We build our tracker upon a pre-trained CNN and remove all fully-connected layers. This special design decreases the model size, and simultaneously increases the speed of online tracking. For example, the proposed tracker can track objects in a speed of over 100 fps on a modern GPU, and at about 30 fps using a single CPU. More importantly, we can train the proposed tracker with less data.

Age and Gender Recognition: In this research, we propose a lightweight deep framework to jointly recognize age and gender in a fast end-to-end manner. The proposed framework employs a multi-task learning scheme to complete these two correlated tasks. Multi-task learning achieves better performance than training a single task one by one. The reason is that multi-task learning can exert a conducive position on extracting the shared feature to improve the accuracy of each task, which can be better than optimize each task. Moreover, it is very easy to extend our model for other more tasks, e.g. facial expression and other attributes. The key contributions of this task are listed as follows:

1) To our knowledge, this is the first attempt to investigate how age and gender recognition can be optimized together to learn a correlated multi-task. Our multi-task learning scheme enables to share and learn optimal features to improve recognition performance for both two tasks. Notably, the proposed model does not limit the number of related tasks, we can extend to many other tasks, e.g. facial expression and other attributes.

2) The network architecture employed by our model is specially designed for age and gender recognition to improve the time efficiency while keeping the quality of recognition performance. Not only outperforming the compared methods [52] in recognition accuracy, the experimental results but also demonstrate that our model runs 10 times faster than [52] and achieves real-time performs even on a low-end PC or a mobile device. Thus, it is very suitable for our model to be implemented in the commercial applications or industry.

1.5 Organization of the Thesis

The remainder of the thesis is organized as follows. In Chapter 2, we propose our depth weight function and convolutional network for object tracking. Then we present our lightweight deep convolutional network in Chapter 3, including the network architecture, multi-task learning scheme, the training/testing procedure of our model, the experimental results, comparisons and component analysis. At last, Chapter 4 concludes this thesis.

Chapter 2

Learning Real time Tracker Using Convolutional Neural Networks

This chapter comprises four sections. The first section briefly introduces the overview of object tracking. Section 2.2 focuses on problem formulation and notations. Section 2.3 focuses on the design of our experiments, including the datasets we adopt and the formulation of performance error measurements. The forth section illustrates the experimental results of our proposed models. The last section is a review of this chapter.

2.1 Overview

As is known to all, object tracking is always a quite interesting and active topic in research community. But many influencing factors, such as diverse poses, different intensity of illumination, complicated backgrounds lead object tracking to be a very tough challenge. Single-target tracking is a core computer vision problem which has attracted many attentions in the past decades [31]. The tracking problem can be considered as a local detection problem within a search region, also called Tracking-by-Detection. To learn a robust target detector, many researchers attempted to model the object's appearance in an online fashion [32], [33]

[12], [15]. Recent successful paradigm for this problem has been exploited the expressive representation power of CNNs. Though, the CNNs have been successfully applied for the problem of visual tracking [21], [22], [23], [24], [25], [26], [27], [28], [29], [30], it is still a challenging task due to the uncertainly changes of object online, such as illumination changes, shape deformation, occlusion, fast motion etc.

Most existing methods focus on either using CNNs as a generic feature extractor [23], [24], [25] or directly training CNN trackers [26], [27]. However, all trackers mentioned above takes no account of the temporal appearance of targets in adjacent frames. Recently, inspired by the success of multiple inputs of CNN model in unsupervised learning [34] and action classification [35], some pair based CNN architecture [28], [29], [30] are employed to address this problem.

Despite achieving state-of-the-art performance, existing pair based CNN trackers still have many drawbacks. Most of these methods utilized two separated CNNs for each input. For example, one CNN stands for target objects [28], [30] or previous cropped regions [29], and the other CNN stream represents the search areas [28], [30] or current cropped regions [29]. However, this strategy results in a noticeable increasing of the number of the model parameters, which consequently requiring more labeled samples at the training stage. Even with multiple inputs, previous works still cannot well model the temporal cues of tracked objects, as they were likely to rely on the networks, which were pre-trained merely on static images [38], and even some of them frozen the network weights for training. Moreover, current CNN trackers can run at over 100 fps on a GPU device, but still execute at very slow

speed on a single CPU processer (around 5 fps in our PC) due to the complexity of their network structure. Therefore, all CNN trackers cannot easily integrated into small devices like mobile phones.

In this thesis, we aim to learn a network in an off-line fashion and track objects online. To make the tracker lighter and incorporate temporal cue, we propose a novel frame-pair based CNN architecture. Specifically, we stack the cropped regions from two successive frames together as the input to a CNN stream. The fusion at early stage allows the tracker directly learn many temporal features [36]. The output of our tracker is a probability map which indicates the location of target. We build our tracker upon a pre-trained CNN and remove all fully-connected layers. This special design decreases the model size, and simultaneously increases the speed of online tracking. For example, the proposed tracker can track objects in a speed of over 100 fps on a modern GPU, and at about 30 fps using a single CPU. More importantly, we can train the proposed tracker with less data.

2.1.1 Depth Information from Kinect

Figure 2.1 Microsoft Kinect

Kinect is a Microsoft product, which is a peripheral for Microsoft's Xbox 360 video game system (Figure 2.1) and a combination of Microsoft's software and PrimeSense's hardware.

Until November 2010, it was called by its codename, “Project Natal.” On November 4, the device was firstly launched as the “Microsoft Kinect” and started to sell. Kinect achieved a big commercial success. It sold around 10 million units in the first month after its first launch, which becomes the fastest selling computer peripheral in the historical records [2]. After its launch date, it is obviously proved that Microsoft’s device Kinect is not only a computer game tool, but also applied for many other applications, such as robotics and virtual reality. Owing to its ability to track movements and voices, and even identify faces, any other additional devices become needless.

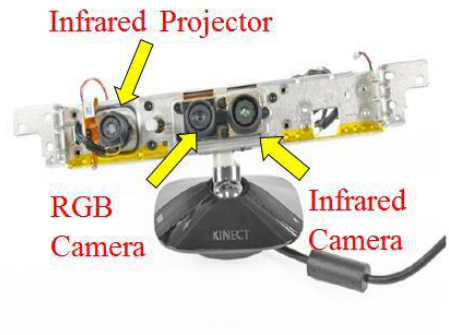

Figure 2.2 A kinect with its plastic casing removed.

Kinect has three “eyes”, as shown in Figure 2.2, from left to right, they are infrared projector, RGB camera and infrared camera, respectively. Infrared projector shines a grid of infrared dots over everything in front of it. These dots are normally invisible to us, but it is possible to capture a picture of them using an infrared camera. Figure 2.3 shows an example of what the dots from the Kinect’s infrared projector look like. Moreover, Kinect can get depth image from infrared camera with resolution of 640*480 pixels (30 fps) shown in Figure 2.4(a). In addition, RGB image can be obtained with the same resolution (30 fps), as displayed in

Figure 2.4(b).

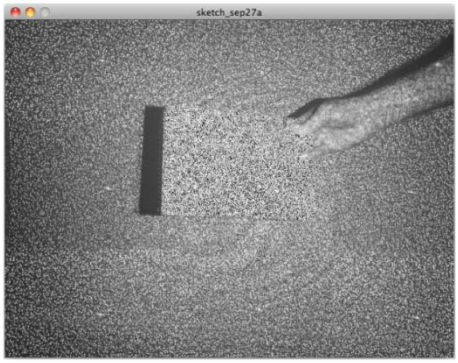

Figure 2.3 An image of normally invisible grid of dots from infrared projector

Figure 2.4 (Left) a depth image, (Right) a RGB image.

The depth image is different from conventional RGB image, whose colour represents not how bright the object is, but how far away it is from the sensor. The brighter parts of the image are closer to camera, and the darker parts are farther away and we can calculate the distance of every object in front of the Kinect.

2.1.2 Compared with traditional RGB images

First of all, a depth image is much easier for a computer to understand than a conventional colour image. Because in a depth image, the colour of each pixel means the distance between the part in the image and the camera. Besides, depth images are not sensitive to the light conditions. Kinect can capture the same depth image in a bright room as in a dark environment, which makes depth images more reliable. In addition, a depth image contains accurate three-dimensional information. Different from a traditional camera, a depth camera can obtain where things are. We can do a lot of funny applications that traditional camera cannot do. For example, we can use the data from a depth camera to reconstruct a 3D model of the object captured. Then we can look at this object in another angle and also can combine it with other 3D models, and even use it to produce new physical objects [2].

2.2 Problem Formulation and Notations

2.2.1 Baseline: Spatio-Temporal Context Learning

In this research, we choose STC [14] as our baseline framework. In STC [14], Zhang et al. apply spatial, temporal and context information for visual tracking. They propose a formulation based on Bayesian theory and low level features to quantify the relationship between the context and the location of target center. Zhang et al. [14] explains that they get the idea of STC method's context prior model from a biological field. In detail, they think if the context location is closer to the center of the target location, this location should have larger weight for estimating the next target location.

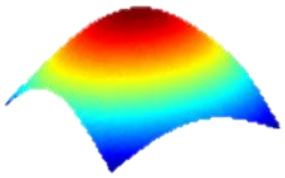

Figure 2.5 Spatial weight function.

The context prior model can be represented as follows:

$$P(c(z)|o) = I(z)\,\omega_\sigma(z - x^*) \tag{2-1}$$

$$\omega_\sigma(z) = a\exp\left(-\frac{|z|^2}{\sigma^2}\right) \tag{2-2}$$

where $I(z)$ is intensity of image, $\omega_\sigma(z)$ is a weight function, a is a normalization parameter and σ is a scale parameter.

In addition, we can describe the confidence map of an object location:

$$c(x) = P(x\,|\,o) = b\exp\left(-\left|\frac{x - x^*}{\alpha}\right|^\beta\right) \tag{2-3}$$

where b is a normalization constant, α is a scale parameter, β is a shape parameter. In STC [14], Zhang set $\beta = 1$ and there is no direct relation between the scale parameter α and the shape parameter β. We can see that spatial, temporal and context information is used in STC method. Even through STC model can obtain good results in its testing datasets, it still has its limitations.

2.2.2 Limitations of Tracking via Spatio-Temporal Context Learning

Even through STC model can obtain good results in its testing datasets, it still has its limitations. We can see that spatial, temporal and context information is used in STC method. However, STC cannot handle the partial occlusion and full occlusion problems. After extensive experiments, we can find that DCF [15], [16] outperforms STC because STC just uses grayscale feature but DCF applies HoG feature. Thus, we will take place raw pixel feature with HoG feature to improve the accuracy of tracking. Moreover, we can see that a CNN could generate from low-level feature to high-level feature in different layers [27]. Besides, we can find that with the help of CNN, researchers have obtained great progress on some high level applications, for example image classification [28] and object detection [29] and etc.

2.3 The Object Tracking Scheme

2.3.1 The Proposed Depth Weight Function

As is listed before, we can observe that depth information can exert a conducive function on ensuring the occlusion situation. Therefore, it is important and profound to implement depth information into improving the tracking accuracy. Motivated by the context prior model and confidence map of STC [14], we propose a depth weight function to assess different level of importance of each section in the following frames by regarding it as the third dimension information. We construct the depth weight function with two constrains:

1) The smaller the distance between the following frames, the more important the section of

this frame will be.

2) If the different distance is large enough, which means there must be an occlusion, thus the priority of this section of the frame should be lower.

Thus, the depth weight function can be formulated as below:

$$w_\lambda(d - d^*) = p \exp\left(-\left| \frac{d - d^*}{\lambda} \right|^\gamma \right) \tag{2-4}$$

where d is current depth of this frame, d^* is depth of last frame, p is a normalization parameter, λ is a scale parameter and γ is a shape parameter (Figure 2.6), there is no direct relation between the scale parameter λ and the shape parameter γ.

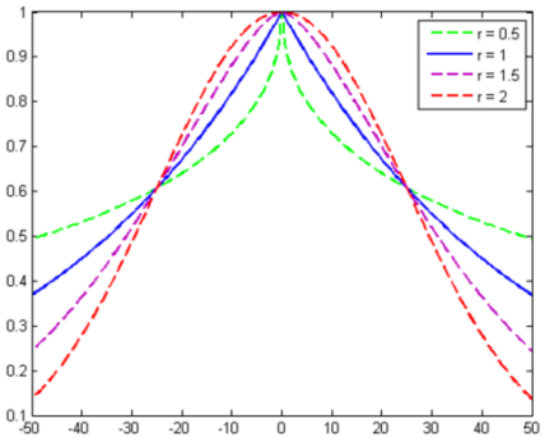

Figure 2.6 Illustration of 1-D depth weight function $w_\lambda(d)$ in (2-4) with different shape parameters.

Finally, bring (2-4) and (2-2) into (2-1), we can propose our depth context prior model, which can be described as follows:

$$P(c(z)\,|\,o) = I(z)w_{\sigma}(z - x^{*})w_{\lambda}(d - d^{*})$$
$$= I(z)a\exp\left(-\frac{\left|z - x^{*}\right|^{2}}{\sigma^{2}}\right)p\exp\left(-\left|\frac{d - d^{*}}{\lambda}\right|^{\gamma}\right) \quad (2\text{-}5)$$

The combination of spatial and depth weight Eq. (2-5) can be described in the Figure 2.7 below:

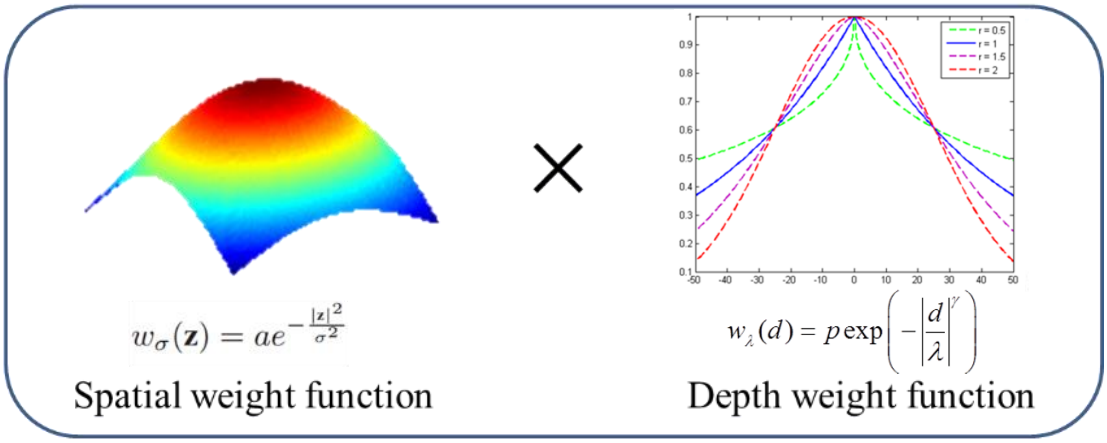

Figure 2.7 The combination of spatial and depth weight function.

With the help of depth information, we can implement the characteristic of spatial, temporal, context and depth information together. Note that, the smaller the distance between the following frames, the more important the section of this frame will be.

2.3.2 The Proposed STC-HoG-CNN

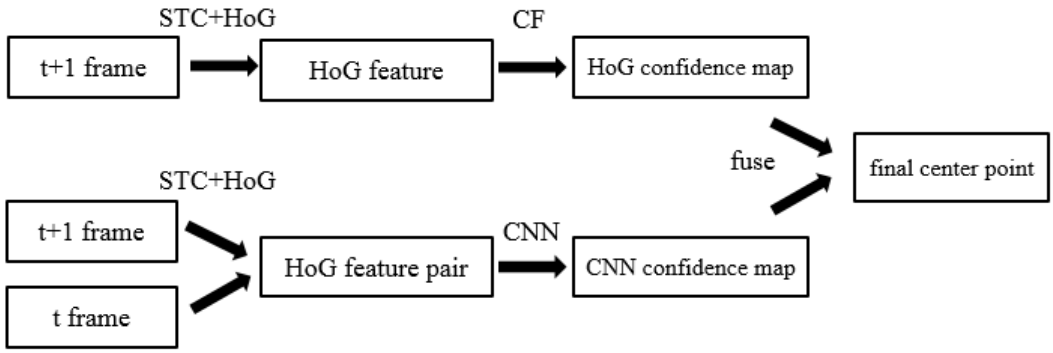

Figure 2.8 Flow chart of the proposed STC-HoG-CNN

As is shown in Figure 2.8, our proposed algorithm has three main steps: pre-training, coarse tracking and fine-grained tracking.

1) Pre-train step: Inspired by STC [14], the CNN network aims to learn a filter from the input HoG feature pair to the output. Then we offline train a CNN network, whose input size is 55×55 and output is a 27×27 confidence map based on the ground truth.

   In the training phase, we choose the TLD dataset [32] and our own dataset (captured by ourselves via Kinect), which totally include 29,301 images and depth images are not used in this dataset. Next, we combine two following frames' HoG feature together and then treat them as a pair. Note that the interval of following frames is 1, 2 and 3, respectively. That is to say, our training dataset is around 90000 pairs with the size of 55×55×62.

2) Coarse tracking: we obtain the HoG feature map by using STC framework and HoG feature. Since the HoG feature map has 31 dimensions, after a linear correlation filter, we can yield the HoG confidence map.

3) Fine-grained tracking: we crop an input region based on the target's center point in last frame and extract HoG feature of this region from the following two frames, which is the input of pre-trained CNN. And then we can obtain the CNN confidence map as the output of CNN. At last, we fuse the confidence map from the first step and this step together to achieve the final confidence map.

$$c_{final} = \phi c_{HoG} + (1 - \phi) c_{CNN} \tag{2-6}$$

where c_{final} is the final confidence map, ϕ is a weight parameter to determine the importance of two confidence maps, i.e. c_{HoG} is the confidence map obtained by HoG feature, c_{CNN} is the confidence map yielded by the CNN part. Finally, we get the prediction center point via the max response of the confidence map c_{final}.

What we are going to deal with are heavy occlusion and even full occlusion problems, and STC and DCF approaches cannot handle them. For example, in Figure 2.9, from #40 to #44, the target bear experiences an entire process: appear - disappear – reappear, which is a very tough task for us.

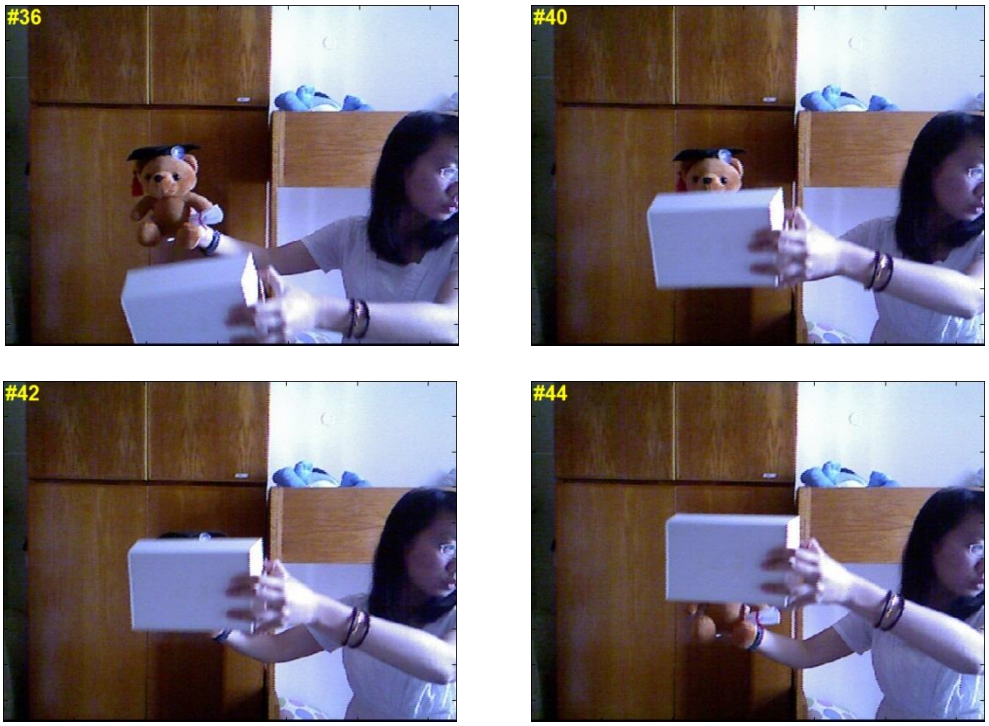

Figure 2.9 The color frames of bear_front frequency of Princeton tracking dataset [37].

2.3.3 The proposed Real-Time Generic Tracker

2.3.3.1 Tracker framework

Our work lies in the frame-pair based tracking framework [28], [29], [30]. Specifically, we are interested in searching a single target object in current frame F_t given the object's location in previous frame F_{t-1} where t is the frame index. Introducing bounding box $B = (x, y, w, h)$ to denote the object location (x, y) and size (x, h), we aim at learning a function to obtain B_t:

$$B_t = \Phi(F_t, F_{t-1}, B_{t-1}) \tag{2-7}$$

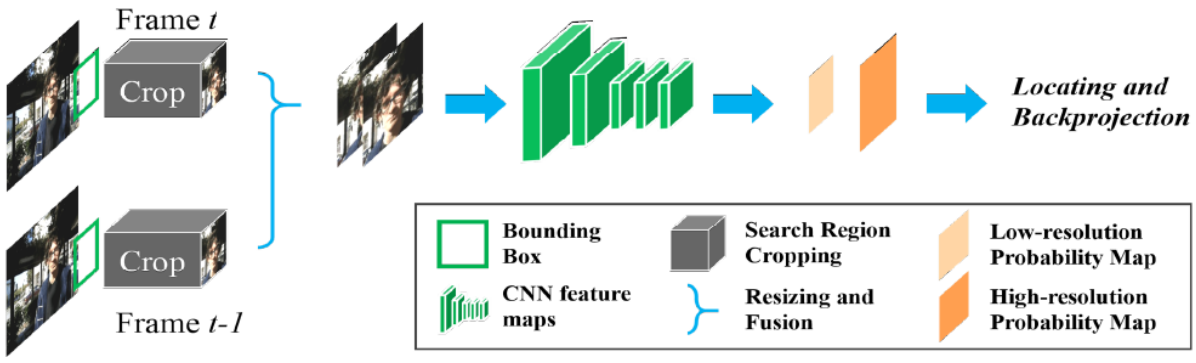

Figure 2.10 Illustration of the proposed tracker. The input to our tracker is a pair of cropped regions and the output of our tracker is a probability map, where the max score indicates the coordinates of tracked target in current frame. We back-project the coordinates to un-cropped original frame to obtain real bounding box. (Best viewed in color and magnification)

Here, we choose the CNN as the function Φ, because CNNs have recently achieved promising results in visual tracking. Our architecture of CNN-based tracker is shown in Figure 2.10, where the input is the concatenation of a pair of cropped regions. The output of our tracker is a probability map where the max score indicates the center of tracked object in the current frame. In detail, we crop a pair of frames (F_t, F_{t-1}) with target center $(x + w/2, y + h/2)_{t-1}$, and size $(kw, kh)_{t-1}$, where $t-1$ stands for the index of previous frame and k defines our search radius. The cropped regions are resized into $m_{in} \times m_{in}$ with scale factor (S_x, S_y), and fed into a CNN model to obtain the l-th convolutional activations. Notably, here we adopt the early fusion technique to make our tracker lighter. Another merit of early fusion at pixel level is that the network can directly learn temporal cues and predict

the motion patterns [36]. Finally, a deconvolutional layer is adopted to upsample the output to finer probability map with size $m_{out} \times m_{out}$, leading to more precision localization [24], [30]. Therefore, each pixel in the final probability map corresponds $r \times r$ region in original network inputs, where $r = m_{in}/m_{out}$ is the sampling factor. However, instead of fixing the interpolation kernel in upsampling, in our experiment, we find that the learned kernel in deconvolution layer locate target better.

Let us briefly compare our tracker to most similar work GOTURN [29]. Our tracker takes full advantage of early fusion (Figure 2.10) and thus noticeable reduces the number of model parameters (from 113M to 9M). Moreover, the output of our tracker is a discrete probability map compared with continuous regression output in [29]. Thanks to above changes, we can train our model with fewer annotated videos, which significant reduce the human efforts for annotating. Finally, our tracker can achieve comparable results with GOTURN and can run at about 30 fps if only a single cpu is available, which is dramatically faster than GOTURN (around 5 fps in our PC).

For employing tracker online, usually only the initial bounding box is given, so that we start from (F_1, B_1), and track target objects in successive frames. For each other frame $\{F_t, t = 2,3, \dots, N\}$, we search the location of target on the output probability map, and project its coordinates back to original un-cropped frame by:

$$\tilde{c} = \frac{c \cdot r}{s} + coord_{crop} \tag{2-8}$$

where $\tilde{c} = (x, y)$ is the center of target bounding box in un-cropped frame, $c = (x, y)$ is coordinates on probability map, $r = (r, r)$ is the sampling factor from the network inputs to

outputs, $s = (S_x, S_y)$ is the resizing factor, and $coord_{crop} = (x, y)$ is the start coordinates of cropped regions in the original frame. All operations in Eq. (2-8) are *element-wise* operations. The B_t can be calculated from $\tilde{c}$ with bounding box size. For simplicity, we fix the size of tracked target across all frames in each video.

2.3.3.2 Tracker design

Our tracker is built upon existing CNNs, which were pre-trained on ImageNet [38] dataset. In order to adapt the pre-trained CNNs to our problem, we take several changes. First, we simply double copy the channels of filters in the first convolutional layer to accept stacked regions. The stacked regions are transformed into feature maps through 5 convolutional layers. Here, we remove all layers after the last pooling layer to preserve more spatial information, and to reduce the model size. We believe smaller network is more suitable for the problem of visual tracking because: 1) visual tracking is a binary classification task that requires much less model complexity than general recognition problems, 2) less parameters make our tracker easy to train with less labeled data. Second, we create another 2 mlp layers on the top of output from the pre-trained CNNs. These two mlp layers help us to learn a more robust tracker. Third, in order to obtain a probability map, a typical operation [39] is to convolute the output of the second mlp layer with a $1 \times 1 \times chns$ filters, where chns is the number of feature maps in the previous layer. In this paper, we adopt an *element-wise cross channel* (EWCC) classifier since it does improve the tracking performance than a typical convolutional layer. We believe the shared weights in a typical convolution operator used in [39] cannot preserve much spatial information as our EWCC layer. Forth, we utilize a

deconvolutional layer to increase the size of probability map to better locate the target object, similar to [39]. Here, we define the EWCC layer as follows:

$$P(i,j) = M(i,j,ch,n) \odot W(i,j,ch) \tag{2-9}$$

where $\odot$ is the *element-wise multiplication*, P is the low resolution probability map, M is the feature maps from the second mlp layer, and W is the classifier weight tensor. P, M, and W are with the same spatial size $Rows \times Cols$. Here, i, j, ch and n are the index for spatial rows, columns, feature channels and sample batch, respectively. For online tracking, n always equals to 1. Now, we show the Backpropagation procedure of new EWCC layer in Eq. (2-10) and Eq. (2-11).

$$\frac{\partial P(i,j)}{\partial M(i,j,ch,n)} = \nabla(i,j,ch) \odot W(i,j,ch) \tag{2-10}$$

$$\frac{\partial P(i,j)}{\partial W(i,j,ch)} = \sum_{n=1}^{N} \nabla(i,j,ch) \odot M(i,j,ch,n) \tag{2-11}$$

Here, Δ is the gradients from successive deconvolutional layer, and N is the training batch size. With above formulations, our tracker can train in an end-to-end manner.

2.3.3.3 Tracker training

We employ logistic loss to train our tracker and define labels on the high-resolution probability map as:

$$y_n = \begin{cases} +1, & if \|u - c\| \leq R/r \\ -1, & if\ otherwise \end{cases} \tag{2-12}$$

where u is the center of bounding box on the probability map. The Eq. (2-12) indicates that on this map, the samples are considered to be positive if they are within radius of R/r. R is set by user before training and fixed across all training epochs. Also, we weight the score map by the positive and negative samples to eliminate class imbalance issue as [30].

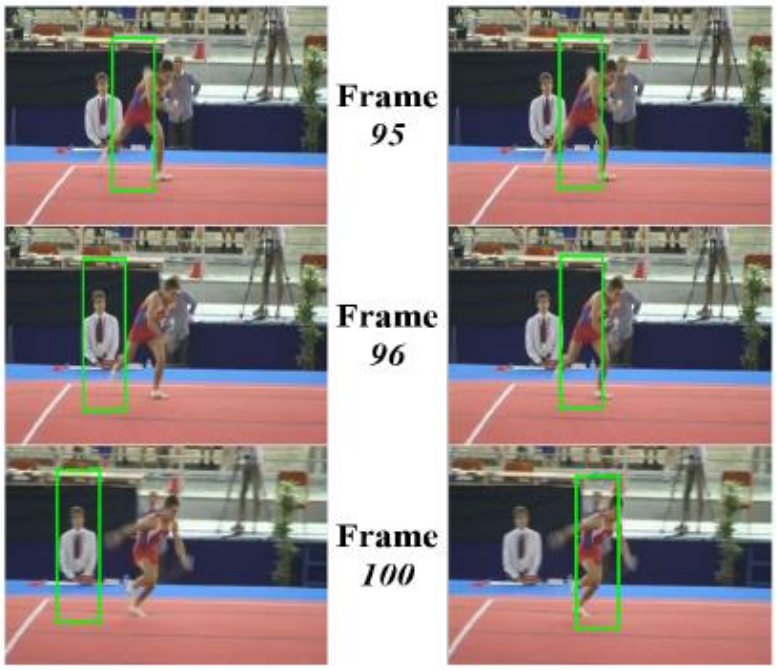

Figure 2.11 Illustration of the tracker trained using different offsets. The results shown in left and right columns are obtained by using the tracker trained under the offset o = 1 and o = 6 respectively (Best viewed in color and magnification).

A set of videos from ALOV300++ [40] is used for tracker training. But in this dataset, the ground truth bounding boxes are just annotated in every 5th frames of each video, we augment the data by generating intermediate bounding boxes for every 5th frames using KCF tracker [15], [16]. The data augmentation can help us learn a more robust tracker. Notably, even by this technique, our training set is still half of data used in [29]. In the training phase, we observe when directly feeding two successive frames always leads tracker failed online, especially in fast motion pattern. To handle this problem, we propose a simple, yet effective sampling strategy to train our tracker. To be specific, in each batch for training, we first sample $N/2$ reference frames $\{t_n, n = 1,2, \dots, N/2\}$, and then get another $N/2$ frames with random numbers ranged between $[t_n - o, t_n + o]$, where o is a hyper-parameter that controls the smoothness of tracking and defined by user. When o is set to 1, the training phase degrades into the one using two successive frames. Figure 2.11 shows the different results obtained using $o = 1$ and $o = 6$ for training and demonstrates the effectiveness of our sampling method.

2.3.3.4 Datasets

Training Set. We train our tracker using a collection of annotated videos. The video sequences come from ALOV300++ [40] dataset. We also remove 7 videos, which are overlapped with our test set, remaining 307 videos for tracker training. However, in this dataset, the ground truth bounding boxes are labeled approximate every 5th frame of each video. We augment the dataset as described in Section 2.3.3.3. After data augmentation, our training set consists of 65,410 images, belonging to 251 different object categories. Note that,

even with our augmentation technique, the total number of images is still less than 147,903 used in GOTURN [29]. Moreover, the ImageNet Detection [38] database utilized in [29] did help their model capture more diverse object appearance. Compared with their work, our tracker still achieves competitive results on the public benchmarks. We split these videos into 250 for training and 57 for validation to fine-tune hyper-parameters. After choosing the hyper-parameters, we retrain our tracker using all 307 videos.

Testing Set. Our goal is to learn a generic tracker, most similar to GOTURN [29]. For a fairly comparison, we evaluate our tracker on the VOT 2014 Challenge database. This database is a popular tracking benchmark that consists of 25 videos in total. We report two popular metrics [31] for evaluations: the Precision and the Area Under the Curve (AUC) for the success plot. We also use a score 20 pixels as threshold for Precision [31].

2.4 Performance Evaluation

The proposed DSTC: In order to test the performance of our proposed depth spatial temporal context (DSTC) model, we use 5 frequencies from Princeton dataset. We compare the proposed DSTC model with other approaches, i.e., Spatial-Temporal Context (STC) [14], Compressive Tracking (CT) [12], [13].

The proposed STC-HoG-CNN: We employ the proposed STC-HoG-CNN method on 51 sequences from Visual Tracking Benchmark (VTB) 1.0 [31] dataset to evaluate our performance with other approaches, i.e. Compressive Tracking (CT) [12], [13], Spatial Temporal Context (STC) [14] and Dual Correlation Filter (DCF) [15]. All the trackers are

evaluated on an i7 3.60 GHz machine with 16GB RAM and only CPU.

The proposed Real-time Generic Tracker. We build our tracker based on VGG-F network [41]. The representation powerful of VGGF is almost the same to AlexNet [7], utilized in GOTURN [29]. For the network, we remove all fully-connected layers and attached two mlp layers on the top of Conv5 activations, followed by a EWCC classifier and a deconvolutional layer.

TABLE 2.1 Illustration of the information of additional layers used in our tracker. All size shapes are formulated as $[h, w, chns, num]$, where h, w, $chns$ and num represent the spatial rows, columns, feature map channels and the filter number, respectively. The word "None" means no "stride" setting in the EWCC layer.

Layer Name	Input Size	Weight Size	Stride	Output Size
MLP1	[13, 13]	[1, 1, 256, 128]	1	[13, 13, 128]
MLP2	[13, 13]	[1, 1, 128, 128]	1	[13, 13, 128]
EWWC	[13, 13]	[1, 1, 128, 1]	None	[13, 13]
DeConv	[13, 13]	[8, 8]	1	[56, 56]

2.4.1 Experiments Setup

The proposed DSTC: In the 5 datasets, we set the same parameters: the scale parameter λ is set to $\lambda = 2.214$ and the shape parameter γ is set to $\gamma = 2$. That is to say, we implement Gaussian distribution to our depth weight function, so as to improve the tracking accuracy efficiently and effectively.

The proposed STC-HoG-CNN: In the formulation (2-6), one parameter needs to be set. After extensive experiments, we set the weight parameter of (2-6) to 0.8. Moreover, we have trained some CNNs to test the performance and finally find the best one is a 4-layer CNN, the input size of which is 55×55×62 and the output size is 27×27.

The proposed Real-time Generic Tracker: For tracker training, to construct a batch for each iteration, we first randomly sample a video, and then select 16 frames using our sampling strategy. The frame offset is 6 for tracker learning. The weights of two MLP and EWCC are generated under a Gaussian distribution, and the weight of deconvolutional (DeConv) layer is initialized with a bilinear kernel. The upsampling factor for DeConv is 4. We execute 50 epochs in total and 1000 iterations per epoch. The optimization is achieved by Stochastic Gradient Descent (SGD) with momentum technique. The learning rate is logarithmically decreased from $log(-2)$ to $log(-4)$. Table 2.1 presents additional layer settings in our tracker, and Table 2.2 shows all parameters used in training phase. After learning, we employed our tracker online without any model updating.

TABLE 2.2 The parameters for trained off-line training.

Frame Offset (o)	Batch Size (N)	Search Radius (k)
6	16	2.5
Positive Radius (R)	Epoch	Iteration Per Epoch
16	50	1000
Learning Rate Range	Momentum Rate	Weight Decay
$[log(2), log(4)]$	0.9	$5e-6$

2.4.2 Performance comparison

We employ success rate (SR) to quantitatively evaluate the different trackers. The score of success rate is defined as $score = \frac{area(R_t \cap R_g)}{area(R_t \cup R_g)}$, where R_t is a tracked bounding box and R_g is the ground truth bounding box, and the result of one frame is considered as a success if score > 0.5.

2.4.3 Quantitative performance comparison

The proposed DSTC

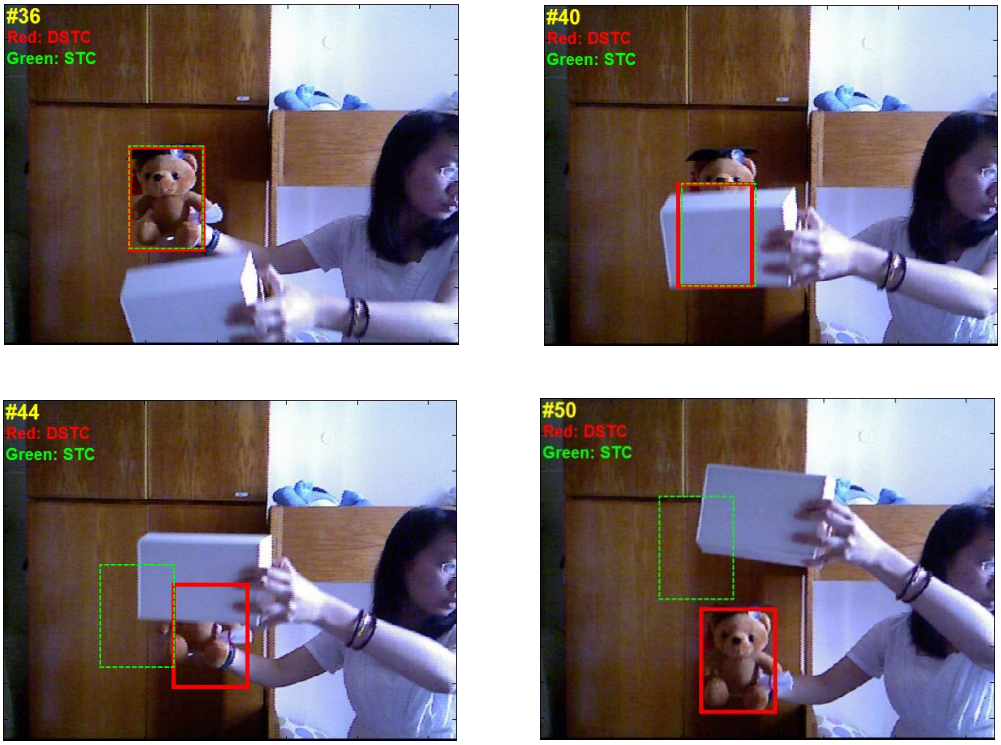

Figure 2.12 Red one is DSTC and green one is STC.

From Figure 2.12, it can be apparently observed that depth information serves a more conducive function in the improvement of tracking accuracy when there is partial and even full occlude in front of our target than STC. We can find that from #40 to #50, the frames

change from partial occlusion to even full occlusion, which is a hard situation we need to tackle. Compared with STC approach, our DSTC can track the target successfully. Besides, Figure 2.13 shows the view of the context region, which can prove that depth information helps to find the location of target.

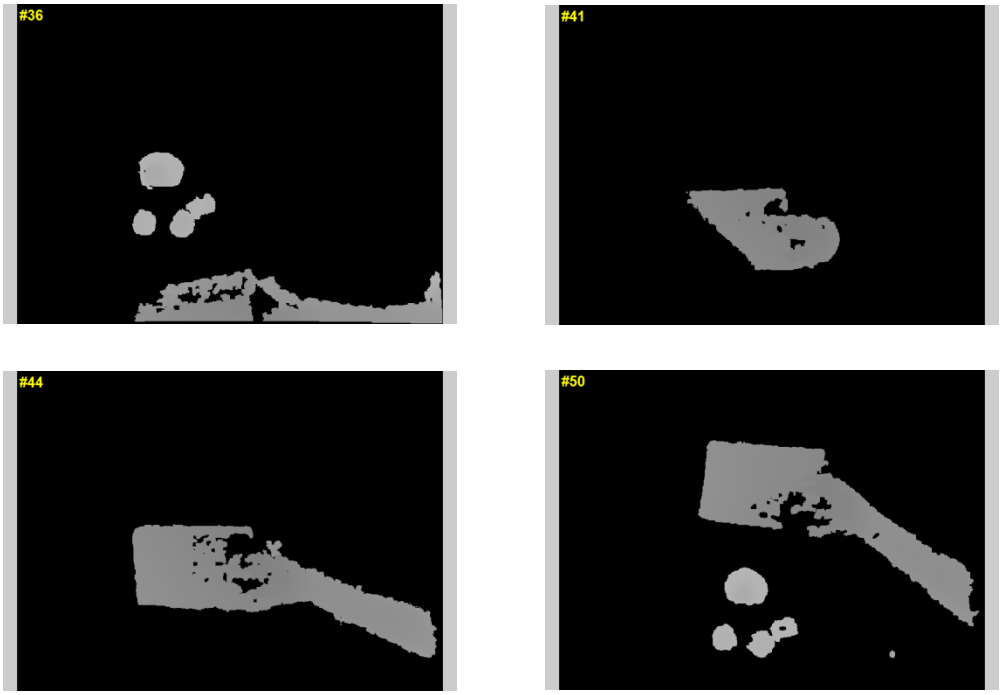

Figure 2.13 Our DSTC method after elimination of the background.

TABLE 2.3 Success rate (SR) (%). **Red** fonts indicate the best performance. The total number of evaluated frames is 1, 210.

Sequence	STC	STC (no scale)	CT	DSTC (no scale)
bear_front	12	14	13	**21**
child_no1	73	98	83	**100**
face_occ5	49	50	50	**50**
new_ex_occ4	49	51	44	**51**
zcup_move_1	84	81	81	**82**
Average SR	54	57	54	**60**

In Table 2.3, it is apparent that DSTC tracker yields best performance than other approaches in SR, i.e., STC [14], CT [12], especially, in bear_front and chilid_no1 sequences, from 13 to 21, and 83 to 100. We observe that in these two sequences, the target is partially occluded and the depth weight function of our DSTC model can help to improve the ability to track the target. While for bear_front and face_occ5 sequences, the accuracy only improves a little or with no help. We find that when the target is fully occluded for a long period, even the depth weight function of DSTC loses the ability to locate the target, due to all the value of depth weight function is 1. Thus, we need to implement an occlusion handling function in our framework to reacquire the target when this situation occurs.

TABLE 2.4 Success rate (SR) (%). Red fonts indicate the best performance.
The total number of evaluated frames is 1, 210.

Sequence	STC	STC (no scale)	CT	DSTC (no scale)	DSTC_OH
bear_front	12	14	13	21	**61**
child_no1	73	98	83	100	**100**
face_occ5	49	50	50	50	**95**
new_ex_occ4	49	51	44	51	**51**
zcup_move_1	**84**	81	81	82	**82**
Average SR	54	57	54	60	**81**

Table 2.4, it is obvious that DSTC_OH obtains best performance than other approaches in success rate. Especially, compared with our DSTC method, DSTC_OH improves significantly in bear_front and face_occ5 sequences, from 21 to 61, and 50 to 95. We observe that in these two sequences, the target is fully occluded for a long period, thus the depth weight function of our DSTC model loses the ability to locate the target, while with the help of occlusion handling mechanism we can find the target again. Therefore, it is necessary to implement an occlusion handling mechanism in our DSTC approach.

The proposed STC-HoG-CNN:

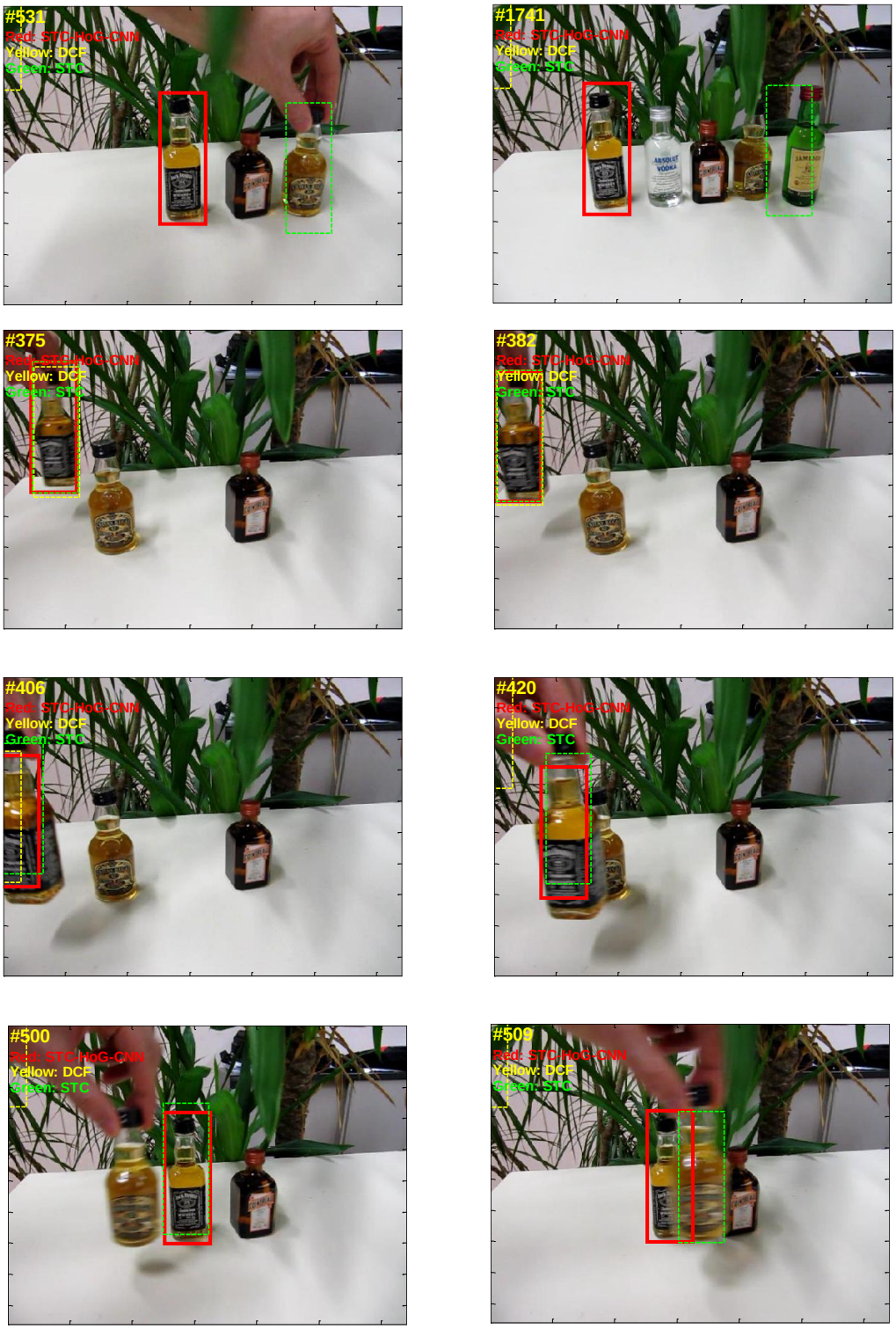

Figure 2.14 Red one is STC-HoG-CNN, yellow one is DCF and green one is STC of Liquor sequence of Visual Tracking Benchmark dataset [31].

TABLE 2.5 Success rate (SR) (%) and frame per second (FPS). Red fonts indicate the best performance. The total number of evaluated frames is 29,491 of VTB 1.0 [31].

Methods	AUC (SR)	Speed (fps)
CT	35.8	106.6
STC	42.1	**199.6**
STC (no scale)	50.4	177.1
DCF	54.7	26.2
DCF (no resize)	63.8	16.4
STC-HoG-CNN	**67.1**	11.1/12.1(GPU)

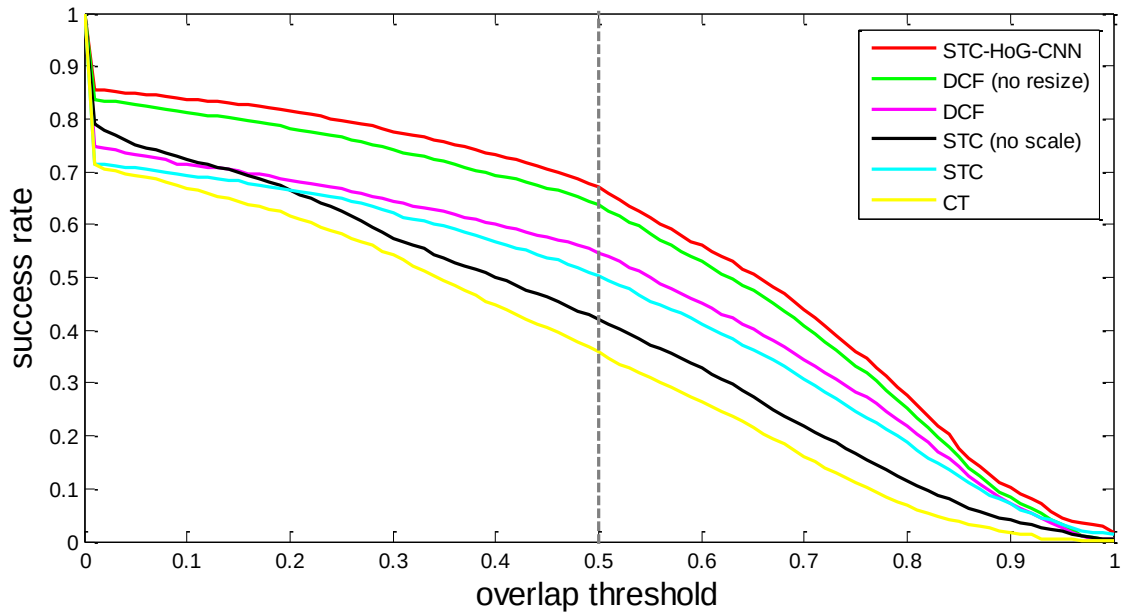

Figure 2.15 Average success rate on different threshold on the VTB dataset.

In Table 2.5, it is obvious that DCF performs better than STC, mainly because the HoG feature used by DCF is much more discriminative than raw pixel employed in STC. Besides, we can notice that our proposed STC-HoG-CNN obtains the best performance, which largely

rises from 63.8 to 67.1 percent compared with DCF (no resize), i.e. 3.3 percent increase. Besides, the speed of our proposed algorithm is almost the same with the second best one, i.e., DCF (no resize). Moreover, if we implement our method on GPU, the speed can be up to 12.1 FPS. That is to say, the proposed method is fast and efficient.

In addition, the success rate on different threshold is shown in Figure 2.15, from which we can see that the proposed STC-HoG-CNN achieve the best performance on the VTB dataset. Moreover, for example, as is shown in Figure 2.14, the target of the Liquor sequence of VTB dataset is a black bottle with yellow liquid. We can notice that after frame #420, DCF loses the target, and after frame #500, STC and DCF both lose the target, but our proposed method can still track the target even when the bottle is occluded with full and very similar occlusion until the end of the sequence (frame #1741). Therefore, we can prove that with the help of CNN, the proposed method can be more effective and efficient than other approaches, i.e. STC and DCF.

The proposed Real-time Generic Tracker:

A series of experiments were carried out to investigate the performance of the proposed tracker. We first present the results obtained by training our tracker under different settings, and then show the comparisons between our method and other related works.

TABLE 2.6 Quantitative results of the comparisons. The experiments are conducted on VOT 2014 dataset. SR and Prec stand for the success rate and precision, respectively. As our goal is to learn a generic object tracker, we only compare the proposed tracker with GOTURN [29], which is also a deep CNN based tracker. All compared methods run by ourself using the public codes. Speeds achieved by CPU and GPU

are marked with C and G.

Compared Models	AUC (SR)	Prec@20	Speed (fps)
KCF	0.421	0.547	C392
DSST	0.401	0.553	C43
GOTURN	0.461	0.610	C5 G153
No Pre-trained CNNs	0.304	0.369	
No Deconv Updating	0.387	0.501	
No EEWC	0.403	0.511	
Full model ($o = 1$)	0.401	0.481	
Full model ($o = 6$)	0.427	0.563	C38 G148

In-depth studies. Table 2.6 shows the results of different model variants. For fairly comparisons, we fix all other settings the same, including the additional layers and the hyperparameters used in optimization. We can find that our full model achieves the best performance among all our variant models. As presented in Table 2.6, the model initialized from a pre-trained CNN contributes most to our performance, which is also demonstrated in other high-level recognition problems [34], [35]. Besides, weight updating in Deconv layer also improves the tracker's performance. In addition, Table 2.6 illustrates our special designed classifier leads to a big improvement and demonstrates that the element-wise cross channel classifier does preserve more spatial information than traditional convolutional layers, which share all spatial cues in its weights [39]. Moreover, the results obtained by our full model with two different frame offset settings are listed in Table 2.6 and demonstrate the importance of our sampling strategy. According to above in-depth analysis, we adopt the following implementations for our tracker in the rest of comparisons, including model

initialized with pre-trained CNNs, DeConv weight updating, EEWC layer, and sampling with frame offset 6.

Comparisons with other methods. In the upper part of Table 2.6, we also show the comparisons to other related works. Our tracker achieves competitive results with other methods. In Table 2.6, our tracker is not as good as the GOTURN [29] both in precision and success rate. A main reason may be the proposed tracker does not model the object's sizes as GOTURN, which is apparently important for visual tracking both in this dataset and in the real-world. We leave this issue for the further work. On the other hand, our tracker only has about 9 million parameters, significant lighter than 113 million in GOTURN. Moreover, the speed of our tracker is approaching the DSST method when running on a CPU device, which is dramatically faster than GOTURN. We believe our proposed tracker is the fastest deep architecture based tracker when performing on a single CPU. In Table 2.6, we do not include any other deep architecture based trackers like [23], [24], [25], [27], because they were not aiming at learning a generic object tracker. In the comparisons to other popular trackers like KCF [15] and DSST [42] our work achieves better performance. Notably, even equipped with deep CNNs, our tracker are just trained in an off-line fashion and do not update its weights online as KCF and DSST. A possible improvement over our proposed tracker is to explore an efficient online update method to adapt the weights to tracked objects.

We also show six of 25 tracking results on several frames. For visualization purposes, all images are stretched to have the same size. The sequences from 1st to 6th rows are: Ball, David, Fish2, Jogging, Sunshade and Woman.

Figure 2.16 Visualization of tracked results using different methods.

TABLE 2.7 Quantitative results of the comparisons. The experiments are conducted on OTB-50 dataset [31]. SR and Prec stand for the success rate and precision, respectively. Speeds achieved by CPU and GPU are marked with C and G.

Compared Models	AUC (SR)	Prec@20	Speed (fps)
GOTURN	0.445	0.646	C5 G153
Full model ($o = 6$)	0.389	0.539	C38 G148

Comparisons with other methods. In Table 2.7, we also show the comparisons to other related works on OTB50 dataset [31]. Our tracker achieves competitive results with other methods. We removed the 9 videos which are overlapped between the two datasets. The results are similar to what we obtained on VOT2014, i.e., our method sacrifices a little the accuracy but achieves much faster speed. Specifically, compared with GOTURN whose AUC (SR) is 0.445 on OTB50, our method has an AUC (SR) of 0.389. This demonstrates that our trained tracker is a generic tracker and can be applied to other datasets. In Table 2.7, our tracker is not as good as the GOTURN [29] both in precision and success rate. A main reason may be the proposed tracker does not model the object's sizes as GOTURN, which is apparently important for visual tracking both in this dataset and in the real-world.

TABLE 2.8 Quantitative results of the comparisons. The experiments are conducted on OTB-100 dataset [31]. SR and Prec stand for the success rate and precision, respectively. Speeds achieved by CPU and GPU are marked with C and G.

Compared Models	AUC (SR)	Prec@20	Speed (fps)
GOTURN	0.422	0.566	C5 G153
Full model ($o = 6$)	0.354	0.444	C38 G148

Comparisons with other methods. In Table 2.8, we also show the comparisons to other related works on OTB100 dataset [31]. We removed the 10 videos which are overlapped between the two datasets. When compared with GOTURN whose AUC (SR) is 0.422 and our method is 0.354. While we can find that our tracker is not as good as GOTURN [29] both in precision and success rate, due to our tracker does not model the object's sizes as GOTURN. In the future, we will focus on this part.

2.5 Summary

The proposed DSTC: In this research, we bring the depth weight function into the tracking problem. This depth weight function sets more weight to the not occluded sections and less weight to the occluded sections. As a result, the proposed depth based algorithm can improve the efficiency and accuracy of tracking especially the occlusion problem. To conclude, I

concede the proposed depth spatio-temporal context (DSTC) model achieves the optimal performance than the other approaches [12], [13], [14]. Additionally, it is necessary to implement an occlusion handling mechanism in our DSTC model.

The proposed STC-HoG-CNN: In this research, firstly we take place raw pixel feature with HoG feature in STC [14] framework. This step is regarded as coarse tracking. Secondly, we bring the Convolution Neural Network (CNN) to learn a filter from the input HoG feature pair to the output. The second step is treated as fine-grained tracking. Finally, we fuse the confidence map obtained by the above two steps to get the final prediction center point of the target. From the comparison results of Visual Tracking Benchmark Dataset, we can see that the proposed STC-HoG-CNN model achieves the best performance than the STC and DCF approaches. To conclude, I concede the proposed STC-HoG-CNN model achieves the optimal performance than the other approaches [12], [14], [15]. Even though the proposed STC-HoG-CNN model achieves quite good results, we still have a lot to improve our tracker's accuracy and efficiency.

The proposed Real-time Generic Tracker: In this research, we present a novel generic tracker framework that can run at a high speed both in GPU and CPU devices. Our tracker is based on recent pair-based inputs CNN frameworks. The proposed tracker is light-weight, achieved by early fusion at image pixel domain. This strategy allows our network to directly learn the temporal appearance of tracked objects. The experiments conducted on public tracking dataset demonstrate the effectiveness of our proposed tracker. Another merits of our method is that the training stage need less annotated videos. We hope our findings could

arouse further researches in general object trackers.

Chapter 3

Learning a Lightweight Convolutional Neural Network for Age and Gender Recognition

This chapter comprises four sections. The first section briefly introduces the development of age recognition, gender recognition and deep CNNs. Section 3.2 focuses on problem formulation and notations. Section 3.3 focuses on the design of our experiments, including the datasets we adopt and the formulation of performance error measurements. The section 3.4 illustrates the experimental results of our proposed models. The last section is a review of this chapter.

3.1 Overview

Real time facial attribute recognition is a very promising and hot topic, especially, recognizing age and gender in a single image has sparked off a great interest in both research community and industrial companies. For the age recognition task, researchers firstly employ the hand-craft local features for representing the distribution of face images, such as Gaussian Mixture Models (GMM) [43], and Hidden-Markov-Model [44] After that, they further presented to use different feature descriptors, for the purpose of representing the face image,

e.g. Gabor feature [45], Biologically-Inspired features (BIF) [46], local binary patterns (LBP) [47], followed by Support Vector Machines. For the task of gender recognition, the research road map is very similar to the development of age recognition, because they are highly correlated tasks and both of them belong to the set of facial attributes. [48] used image intensities followed by SVM classifiers and [49] implemented Webers' Local texture Descriptor [50], both of which are based on human hand-craft features.

More recently, the popular deep learning technique achieves incredible progress in visual recognition [51], and also has been successfully applied to age and gender recognition. Levi et al. [52] proposed to individually train two models for each problem, which can be treated as a cascade of CNN. However, these CNN based methods are time consuming for mobiles or low-end PCs for the following two issues:

1) Exploiting the complex CNN architecture. Most of CNN based methods directly employ the popular architectures (say AlexNet [51] and VGG [53]), which are specially designed for large scale visual recognition, e.g. ImageNet Challenge [54]. The network architectures are very complex and overdesigned for the age and gender recognition, which heavily increases the computation burden.

2) Regarding age and gender recognition as two independent problems. Although in [52] the model they trained for age and gender recognition has a same architecture, the parameters are different and the method requires a complex cascade architecture of deep model. As the matter of fact, age and gender recognition are two highly correlated tasks about facial attributes. It will be beneficial to recognize accuracy and time efficiency if we can

optimize these two tasks together.

To address above mentioned issues, we propose a lightweight deep framework to jointly recognize age and gender in a fast end-to-end manner. The proposed framework employs a multi-task learning scheme to complete these two correlated tasks. As is known to all, the effectiveness of multi-task learning has been verified on many computer vision problems, e.g. image classification [55], visual tracking [56] and facial landmark detection [57]. On these problems, multi-task learning achieves better performance than training a single task one by one. The reason is that multi-task learning can exert a conducive position on extracting the shared feature to improve the accuracy of each task, which can be better than optimize each task. Moreover, it is very easy to extend our model for other more tasks, e.g. facial expression and other attributes.

The key contributions of this paper are listed as follows: 1) To our knowledge, this is the first attempt to investigate how age and gender recognition can be optimized together to learn a correlated multi-task. Our multi-task learning scheme enables to share and learn optimal features to improve recognition performance for both two tasks. Notably, the proposed model does not limit the number of related tasks, we can extend to many other tasks, e.g. facial expression and other attributes. 2) The network architecture employed by our model is specially designed for age and gender recognition to improve the time efficiency while keeping the quality of recognition performance. Not only outperforming the compared methods [52], [58] in recognition accuracy, the experimental results but also demonstrate that our model runs 10 times faster than [52] and achieves real-time performs even on a low-end

PC or a mobile device. Thus, it is very suitable for our model to be implemented in the commercial applications or industry.

3.2 Problem Formulation and Notations

In this section, we will present our proposed lightweight deep convolution neural networks for Age and Gender recognition. The framework of our model consists of two stages: shared feature extraction and multi-task estimation stage. In the following, we will describe our model from two aspects: Network architecture and multitask learning scheme.

3.3 The Gender and Age Recognition Scheme

3.3.1 Network Architecture

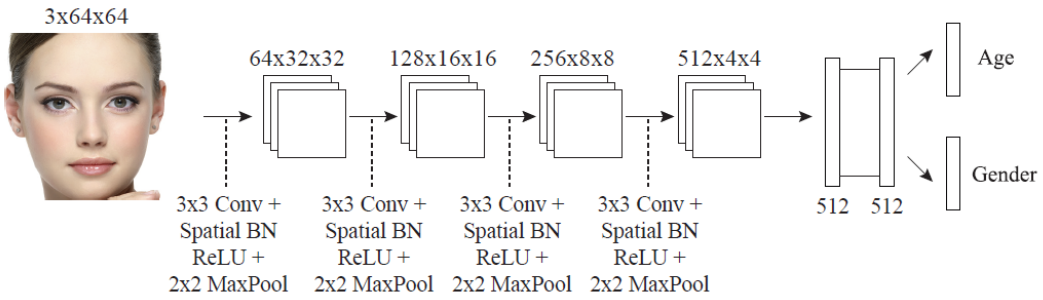

Figure 3.1 The architecture of our lightweight deep model for the age and gender recognition. The network consists of four convolution operations and two fully connected layers, where the raw image pixel are treated as the input.

As illustrated in Figure 3.1, our proposed model is constructed by stacking four special

convolution operations and two fully connected layers. The convolution operation in this work includes Convolution (Conv) + Spatial Batch Normalization (Spatial BN) + Rectified Linear Unit (ReLU) + Max Pooling (MaxPool). The number of convolution filters for the four convolution operations are 64 with $3 \times 3 \times 3$ size, 128 with $64 \times 3 \times 3$ size, 256 with $128 \times 3 \times 3$size, 512 with $128 \times 3 \times 3$size, respectively. The two fully connected layers both have 512 neurons. The input RGB image is downsampled into the size 64×64 before being fed into the network. Notably, our experiments demonstrate that 64×64 resolution is good enough for age and gender recognition tasks. Then the network regresses 2-dimension vector with the estimated age and gender labels for the input image. In summary, our network architecture supports low resolution of the input image and consists of small convolutional filters and fully connectedly layers. This implies that our proposed network is lightweight.

Thanks to the lightweight design, our network architecture has only 6×10^6 parameters and is significantly lightweight, compared with the popular architecture AlexNet [51] (60×10^6 parameters, $10 \times$ bigger than us) and VGG-16 [53] (138×10^6 parameters, $23 \times$ bigger than us). That is to say, our proposed method is simple but effective, and have great impact on solving the time consuming problem. Hence, our network is able to be implemented in a low-end PC or even a mobile device.

3.3.2 Multi-task Learning Scheme

In order to jointly perform age and gender recognition, we exploit the multi-task learning scheme by regarding these two correlated tasks as a regression problem. Specifically, built

upon the last fully connected layer, the output of our model is a regressed label vector with two prediction results for age and gender, respectively. In this way, both the age and gender tasks share the same feature representation. The goal of our network is to learn the shared feature to complete these two correlated tasks. Many research [55], [56], [57] clarify that the multi-task learning scheme is able to improve the generalization performance of multiple related tasks. In the following section, we will describe the formulation of our exploited multi-task learning scheme.

Suppose we have N training samples, C tasks ($C = 2$ in this work) to be completed and y_i^c is the ground truth label of the c-th task for the i-th image I_i. Thus the objective of our proposed multi-task learning scheme is defined as:

$$\arg \min_{\{\omega, \mathrm{w}_c\}} \sum_{i=1}^{N} \sum_{c=1}^{C} l(y_i^c, \phi(I_i, \omega); \mathrm{w}_c) + \Psi(\mathrm{w}_c), \tag{3-1}$$

where $\phi(I_i, \omega)$ denotes the feature vector of our model, ω is the corresponding network parameter. w_c is the regression parameter for the c-th task. The w_c is the L_2 norm regularization term that penalizes the complexity of w_c to avoid model overfitting, i.e., $\Psi(\mathrm{w}_c) = \|\mathrm{w}_c\|_2^2$. The function $l(\cdot,\cdot)$ denotes the estimation error for label regression, and is defined as follows:

$$l(y_i^c, \phi(I_i, \omega); \mathrm{w}_c) = \|y_i^c - \mathrm{w}_c^T \phi(I_i, \omega)\|_2^2. \tag{3-2}$$

To this end, we have presented our deep model for age and gender recognition. Our model has a lightweight architecture and jointly optimizes these two tasks by learning a shared feature representation for regression. In the next subsection, we will discuss the training and

testing phase of our proposed model.

3.3.3 Model Training and Testing

As our proposed model regards the age and gender recognition task as a regression formulation, the standard back propagation algorithm [36] is applicable to optimize the model parameters $\{w, \{\mathrm{w}_c\}_{c=1}^{C}$. Specifically, the partial derivatives with respect to $\{w, \{\mathrm{w}_c\}_{c=1}^{C}\}$ are defined as:

$$\partial \frac{l(y_i^c, \phi(I_i, \omega); \mathrm{w}_c)}{\omega} = 2\left(y_i^c - \mathrm{w}_c^T \phi(I_i, \omega)\right) \partial \frac{\phi(I_i, \omega)}{\omega}$$

$$\partial \frac{l\left(y_i^c, \phi(I_i, \omega); \mathrm{w}_c\right)}{\mathrm{w}_c} = -2\left(y_i^c - \mathrm{w}_c^T \phi(I_i, \omega)\right) \phi(I_i, \omega) \tag{3-3}$$

Once all of above mentioned derivatives are obtained, we can perform the stochastic gradient descending method to update the model parameters $\{w, \{\mathrm{w}_c\}_{c=1}^{C}\}$. In the testing phase, given an input image, our model can directly output both the age and gender estimation result through forwarding the network.

3.4 Performance Evaluation

3.4.1 Dataset Description and Setting

To verify the effectiveness and efficiency of our proposed model, we conduct experiments on the recently released Adience benchmark [79], which is mainly constructed for age and gender recognition and contains 26k images of 2,284 different people with the resolution

816×816. The Adience benchmark is very challenge because its images are directly collected from mobile devices. As illustrated in Figure 3.2, these images are highly unconstrained with extreme variations in head pose, lightning conditions quality, which can represent the real-world challenge. There are 8 categories (i.e., 0-2, 4-6, 8-13, 15-20, 25-32, 38-43, 48-53, 60-) to represent the age level of the subjects in Adience benchmark.

Figure 3.2 Example face images for age and gender recognition from the Adience benchmark.

Our model is trained from scratch and no outside data are used in training phase. Stochastic gradient decent (SGD) is employed to optimize the parameters with image batch size of 128 images. The initial learning rate is 0.1, reduced to one tenth after 25 iterations. The momentum is 0.9. The weight decay is 5×10^{-4}. The learning rate decay is 10^{-7}. Dropping out strategy is also used in the fully connected layers with 0.5 ratio. Training the network on the Adience benchmark takes around 1 hour. We compare our model with three state-of-the-art methods [49], [52], [58] from accuracy and efficiency. Adopting the

evaluation protocol of [52], we perform a standard five-fold, subject-exclusive cross-validation to obtain the recognition accuracy of age and gender. Recognition accuracy is defined as the number of corrected classified samples divides total sample number.

3.4.2 Comparison Results

Recognition Accuracy: Table 3.1 illustrates the recognition accuracy on the Adience benchmark. As is reported in Table 3.1, the prediction accuracy of [49], [52], [58] and our model for gender recognition are 77.8%, 79.3%, 85.9% and 86.0%, respectively. This result demonstrates that our model can obtain the comparable performance with the state-of-the-art methods. Some correctly predicted samples and false estimations are illustrates in Fig. 3.3.

Figure 3.3 The age and gender estimations of our proposed model. The samples in the first and second rows demonstrate that the age and gender are correctly predicted in black, while the last row shows failure cases with wrong prediction in red.

TABLE 3.1 Comparison of average gender and age recognition results. The entries with best accuracy are bold-faced.

Method	Gender	Age
Sun et al. [58]	77.8 ± 1.3	45.1 ± 2.6
Ullah et al. [49]	79.3 ± 0.0	-
Levi et al. [52]	85.9 ± 1.4	$\mathbf{49.5} \pm 4.4$
Ours	$\mathbf{86.0} \pm 1.2$	47.2 ± 1.1

Failure Analysis: In Fig. 3.3, we show some correctly predicted and false estimations samples and the third row illustrates the wrong prediction samples. In the experiments, we find that the first image in the third row we predict the age level of girl's face to be 4-6, but the ground truth is 0-2. While we think our result should be more reliable and closer to her appearance age. For the second image, our output is female with age level 8-12, while actually there are two faces in the photo, the ground truth is male with 25-32, which should be the father's facial attributes. For the forth image, our prediction is age level 38-43, but the correct label is 25-32, we think this person looks older than her physical age. For the fifth image, it is very dark and our method makes the result older than her real age from 25-32 to 48-53. For the eighth photo, the woman looks downside and we consider the appearance age much younger from 25-32 to 8-12. In the experiments, we find: 1) our method does not work well for child and older age level, which results from these two groups have similar appearance features. 2) The illumination situation could also influence the recognition accuracy, e.g. the fifth image. 3) The pose of the face may cause the result to be wrong.

TABLE 3.2 Comparison of average gender and age recognition results with different model structures. The entries with best accuracy are bold-faced.

Method	Gender	Age
Resnet + No Pooling	70.3	32.1
Resnet + Pooling	71.0	33.2
Ours (2fc - half)	77.8	45.8
Ours (1fc)	82.7	46.1
Ours	**86.0**	**47.2**

Detail Comparison: In order to compare different model structures for our two task regression problem, we conduct the following experiments. In table 3.2, we can see from the results that when we remove last fully connected layer Ours ("1fc"), the performance decreases from 47.2 to 46.1, which indicates that last fully connected layer helps to improve the performance. Besides, we want to find whether convolutional filter numbers influence the performance, thus we compare the same structure with only half of the parameters Ours ("2fc – half"), i.e. the convolutional filter numbers is half of Ours "2fc". From the experiments, we can demonstrate when the parameters is half, the result decreases from 47.2 to 45.8. Moreover, we also compare with the recent popular network Resent-20 [80], which is specially designed for Cifar 10 dataset (input size is 32×32). Because the input size of our photo is 64×64, we modify the Resnet structure as following ways: 1) "Resnet + Pooling": we add one more convolution operation (filters are with $16 \times 3 \times 3$ size) as the first layer

and one more max pooling layer, in order to keep other structure the same as resenet-20 for Cifar-10 dataset. 2) “Resnet + No Pooling”: we set the first convolution layer filters with $16 \times 7 \times 7$ size, stride is 2 and padding is 3, thus there is no need to include another max pooling layer. Compared with these two modifications, we find the first one is better, while it only obtain around s30% accuracy for age recognition. We find the two Resnet models meet the over-fitting issue, the training accuracy is almost 80%, while the testing performance only increases to around 30%.

Time efficiency: Firstly, we compare the running time of ours with several CNN architectures, e.g., AlexNet [51], VGG [53], [52], for age and gender estimation on PC platform, which is a desktop with intel 4.0 Ghz CPU and nvidia 970 GPU. Given an input image, the average running times are illustrated in the first row of Table 3.3. It is obvious that our model is $15\times$, $45\times$, $9\times$ faster than the compared AlexNet [51], VGG[53], [52] on the PC platform, respectively.

TABLE 3.3 Comparison of the average running time (second per image) for different network architecture.

Architecture	AlexNet [51]	VGG [53]	Levi et al. [52]	Ours
PC	0.15	0.45	0.09	0.01
Mobile	inapplicable	inapplicable	5.2	0.5

Meanwhile, considering age and gender recognition are very promising applications in

mobile platform, we choose the fastest two models, i.e., [52] and ours to implement on a Samsung Note 3 mobile phone, which has very limited computation capability. As for a mobile application, the processing time is a key factor for users. We can see from the second row of Table 3.3, though obtaining slightly better performance, the compared method [52] requires about 5 seconds for a single image. This processing time is not acceptable. Thanks to the lightweight advantage, Table 3.3 demonstrates that our model only costs 0.5s to predict age and gender for a single image, and is about 10 times faster than [52]. The speed advance is due to that our proposed model has only a few parameters and supports low-solution images. Hence, our model significantly outperforms [52] on the running time.

From the aspect of effectiveness and efficiency, the experimental results validate the contribution of our network architecture. Thanks to the lightweight design, our model is able to be applied to common mobile devices.

TABLE 3.4 Component analysis of single and multiple task with different number of convolution operations

Method	Gender	Age
Ours (single-6conv)	85.3 $\pm$ 0.8	$\mathbf{49.7}$ $\pm$ 0.6
Ours (single-5conv)	85.0 $\pm$ 1.0	49.0 $\pm$ 0.5
Ours (single-4conv)	84.4 $\pm$ 0.5	48.3 $\pm$ 0.7
Ours (single-3conv)	83.7 $\pm$ 0.7	47.6 $\pm$ 0.8
Ours	$\mathbf{86.0}$ $\pm$ 1.2	47.2 $\pm$ 1.1

To specify the contribution of our employed multi-task scheme, we have conducted the following experiments. We construct the single task version of our model, i.e., we discard the multi-task scheme and separately train our model for gender recognition. Moreover, we also investigate that the performance of convolution layer number from 3 to 6. We denote them as Ours ("single-3conv", "single-4conv", "single-5conv" and "single-6conv"), respectively. Notably, Ours "single-4conv" has the same architecture of our multi-task model, while "single-5conv" and "single-6conv" has one and two more convolution operations (filters are 512 with $512 \times 3 \times 3$ size). The experiment results are illustrated in Table 3.4, where we can obtain two observations: i) As the network becomes deeper, the gender recognition accuracy increases from 83.7% to 85.3%. This meets the "the deeper, the better" conclusion; ii) With only 4 convolution layers, our model achieves the best performance on the gender recognition. This justifies the effectiveness of the employed multi-task scheme for the gender recognition. The multi-task scheme can simplify the CNN architecture and improve the performance together.

3.5 Summary

Recently, many CNN based methods directly implement the popular architectures (AlexNet [51] and VGG [53]). For the case of age and gender recognition tasks, however, these network architectures are too complex and over-designed. Moreover, age recognition and gender recognition are two highly correlated tasks about facial attributes. It will be beneficial to improving recognition accuracy and efficiency if we can optimize these two tasks together.

In this paper, we investigated how age and gender recognition can be optimized jointly via a lightweight deep model. Not only obtaining competitive performance with the state-of-the-art methods, our proposed approach also runs much faster. Moreover, thanks to the proposed multi-task learning scheme, our model can be easily extended to other facial attribute recognition tasks, e.g., facial expression, face recognition and facial similarity.

Chapter 4

Conclusions and Future Works

4.1 Conclusions

In this thesis, we mainly focus on two tasks in computer vision applications: object tracking and facial attribute analysis, especially for age and gender recognition. For the object tracking application, we put forward three trackers, i.e. Depth Spatio-Temporal Context (DSTC) tracker, STC-HoG-CNN tracker and Real-Time Generic Tracker (RTGT). DSCT tracker improves the accuracy with the help of depth information, while the other two CNN based tackers use directly from RGB images and may implement depth images for training and testing in the future. From DSCT tracker, we find that depth information is suitable to deal with the occlusion situations, which lead the tacker more robust in a challenging dataset.

For facial attribute analysis application, we propose a lightweight CNN based model for age and gender recognition. In this study, we find that current network architectures are too complex and over-designed, thus we specially design a lightweight deep model for these recognition tasks. Moreover, since these two recognition tasks are two highly correlated tasks, it will be beneficial to improving recognition accuracy and efficiency if we can optimize

these two tasks together.

Object Tracking:

Firstly, in this thesis, we propose a Depth Spatio-Temporal Context (DSTC) tracker, which brings the depth weight function into the tracking problem. This depth weight function sets more weight to the not occluded sections and less weight to the occluded parts. As a result, the proposed depth information can help to improve the efficiency and accuracy of visual tracking especially the occlusion situation. Therefore, I concede the proposed DSTC model achieves the optimal performance than the other approaches [12], [13], [14].

Secondly, we propose a STC-HoG-CNN tracker, which brings the CNN to visual tracking problem: 1) we take place raw pixel feature with HoG feature in STC [14] framework. This step is regarded as coarse tracking. 2) We bring the CNN to learn a filter from the input HoG feature pair to the output. The second step is treated as fine-grained tracking. 3) We fuse the confidence map obtained by the above two steps to get the final prediction center point of the target. From the comparison results of Visual Tracking Benchmark Dataset, we can see that the proposed STC-HoG-CNN model achieves the best performance than the STC and DCF approaches. To conclude, I concede the proposed STC-HoG-CNN model achieves the optimal performance than the other approaches [12], [14], [15].

Thirdly, we present a novel generic tracker framework that can run at a high speed both in GPU and CPU devices. Our tracker is based on recent pair-based inputs CNN frameworks. The proposed tracker is light-weight, achieved by early fusion at image pixel domain. This strategy allows our network to directly learn the temporal appearance of tracked objects. The

experiments conducted on public tracking dataset demonstrate the effectiveness of our proposed tracker. Another merits of our method is that the training stage need less annotated videos. We hope our findings could arouse further researches in general object trackers.

Age and Gender Recognition: Recently, many CNN based methods directly implement the popular architectures (AlexNet [51] and VGG [53]). However, these network architectures are too complex and over-designed. Moreover, age recognition and gender recognition are two highly correlated tasks about facial attributes. It will be beneficial to improving recognition accuracy and efficiency if we can optimize these two tasks together. In this paper, we investigated how age and gender recognition can be optimized jointly via a lightweight deep model. Not only obtaining competitive performance with the state-of-the-art methods, our proposed approach also runs much faster. Moreover, thanks to the proposed multi-task learning scheme, our model can be easily extended to other facial attribute recognition tasks, e.g., facial expression, face recognition and facial similarity.

4.2 Future Works

Object Tracking: Even though the proposed STC-HoG-CNN model achieves quite good results, we still have a lot to improve our tracker’s accuracy and efficiency. Therefore, in the next step, we will do as follows:

1) We will train other CNN architectures to evaluate their tracking performance, for the purpose of getting an optimal CNN.

2) We will fuse the HoG confidence map and CNN confidence map in a more reliable way, in order to balance the performance of HoG and CNN feature.

3) We can replace with linear correlation to Kernelized Correlation Filter (KCF) to evaluate its performance.

Age and Gender Recognition: Thanks to the proposed multi-task learning scheme, our model can be easily extended to other facial attribute recognition tasks, e.g., facial expression, face recognition and facial similarity.

Bibliography

[1] L. Xia, C. C. Chen, J. K. Aggarwal, “Human Detection Using Depth Information by Kinect,” Proc. IEEE Int’l Conf. Computer Vision and Pattern Recognition Workshops, 2011.

[2] G. Borenstein, Making Things See: 3D vision with Kinect, Processing, Arduino, and MakerBot, O'Reilly Media, 2012.

[3] N. Dalal, B. Trigges, “Histograms of oriented gradients for human detection,” Proc. IEEE Int’l Conf. Computer Vision and Pattern Recognition, 2005.

[4] D. G. Lowe, “Object Recognition from Local Scale-Invariant Features,” Proc. IEEE Int’l Conf. Computer Vision, 1999.

[5] B. C. Vemuri, A. Mitiche, “Curvature-based Representation of Objects from Range Data”, Image and Vision Computing, Vol. 4, No. 2, pp. 107-114, May 1986.

[6] M. Andriluka, S. Roth, “People-tracking-by-detection and people-detection-by-tracking,” Proc. IEEE Int’l Conf. Computer Vision and Pattern Recognition, 2008.

[7] D. Ramanan, D. Forsyth, A. Zisserman, “Strike a pose: Tracking people by finding stylized poses,” Proc. IEEE Int’l Conf. Computer Vision and Pattern Recognition, 2005.

[8] P. Viola, M. Jones, “Rapid object detection using a boosted cascade of simple features,” Proc. IEEE Int’l Conf. Computer Vision and Pattern Recognition, 2001.

[9] M. Ozuysal, V. Lepetit, F. Fleuret, P. Fua, “Feature harvesting for tracking-by-detection.

Proc. IEEE Int'l European Conference on Computer Vision, 2006.

[10] C. Rosenberg, M. Hebert, H. Schneiderman, "Semi-supervised self-training of object detection models. IEEE Workshops on Application of Computer Vision, 2005.

[11] P. Roth, M. Donoser, H. Bischof, "On-line learning of unknown hand held objects via tracking," Proc. IEEE Int'l Conf. Computer Vision Systems, 2006.

[12] K. H. Zhang, L. Zhang, M. H. Yang, "Real-time Compressive Tracking," Proc. IEEE Int'l European Conference on Computer Vision, 2012.

[13] K. Zhang, L. Zhang, and M. Yang, "Fast Compressive Tracking," IEEE Transaction on Pattern Analysis and Machine Intelligence, 2014.

[14] K. Zhang, L. Zhang, M-H. Yang, Q. Liu, and D. Zhang, "Fast Tracking via Dense Spatio-Temporal Context Learning," Proc. IEEE Int'l European Conference on Computer Vision, 2014.

[15] J. F. Henriques, R. Caseiro, P. Martins, J. Batista, "Exploiting the Circulant Structure of Tracking-by-detection with Kernels," ECCV - European Conference on Computer Vision, 2012.

[16] J. F. Henriques, R. Caseiro, P. Martins, J. Batista, "High-Speed Tracking with Kernelized Correlation Filters," IEEE Transaction on Pattern Analysis and Machine Intelligence, 2015.

[17] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton, "Imagenet classification with

deep convolutional neural networks," in NIPS, 2012, pp. 1097–1105.

[18] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in ICLR, 2015.

[19] Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell, "Decaf: A deep convolutional activation feature for generic visual recognition.," in ICML, 2014, pp. 647–655.

[20] Ali Sharif Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson, "Cnn features off-the-shelf: an astounding baseline for recognition," in CVPRW, 2014, pp. 806–813.

[21] Hanxi Li, Yi Li, and Fatih Porikli, "Robust online visual tracking with a single convolutional neural network," in ACCV. Springer, 2014, pp. 194–209.

[22] Hao Guan, Xiangyang Xue, and An Zhiyong, "Online video tracking using collaborative convolutional networks," in ICME. IEEE, 2016, pp. 1–6.

[23] Martin Danelljan, Gustav Hager, Fahad Shahbaz Khan, and Michael Felsberg, "Convolutional features for correlation filter based visual tracking," in ICCVW, 2015, pp. 58–66.

[24] Chao Ma, Jia-Bin Huang, Xiaokang Yang, and Ming-Hsuan Yang, "Hierarchical convolutional features for visual tracking," in ICCV, 2015, pp. 3074–3082.

[25] Yuankai Qi, Shengping Zhang, Lei Qin, Hongxun Yao, Qingming Huang, and Jongwoo

Lim Ming-Hsuan Yang, “Hedged deep tracking,” in CVPR, 2016.

[26] Lijun Wang, Wanli Ouyang, XiaogangWang, and Huchuan Lu, "Visual tracking with fully convolutional networks,” in ICCV, 2015, pp. 3119–3127.

[27] Hyeonseob Nam and Bohyung Han, “Learning multi-domain convolutional neural networks for visual tracking,” in CVPR, 2016.

[28] Ran Tao, Efstratios Gavves, and Arnold W M Smeulders, “Siamese instance search for tracking,” in CVPR, 2016.

[29] David Held, Sebastian Thrun, and Silvio Savarese, “Learning to track at 100 fps with deep regression networks,” in ECCV, 2016.

[30] Luca Bertinetto, Jack Valmadre, Joao F Henriques, Andrea Vedaldi, and Philip HS Torr, “Fully-convolutional siamese networks for object tracking,” in ECCV. Springer, 2016, pp. 850–865.

[31] Yi Wu, Jongwoo Lim, and Ming-Hsuan Yang, “Online object tracking: A benchmark,” in CVPR, 2013, pp. 2411–2418.

[32] Zdenek Kalal, Krystian Mikolajczyk, and Jiri Matas, “Tracking-learning-detection,” PAMI, vol. 34, no. 7, pp. 1409–1422, 2012.

[33] Sam Hare, Amir Saffari, and Philip HS Torr, “Struck: Structured output tracking with kernels,” in ICCV. IEEE, 2011, pp.263–270.

[34] Carl Doersch, Abhinav Gupta, and Alexei A Efros, “Unsupervised visual representation

learning by context prediction," in ICCV, 2015, pp. 1422–1430.

[35] Karen Simonyan and Andrew Zisserman, "Two-stream convolutional networks for action recognition in videos," in NIPS, 2014, pp. 568–576.

[36] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei, "Large-scale video classification with convolutional neural networks," in CVPR, 2014, pp. 1725–1732.

[37] S. Song, J. Xiao, "Tracking revisited using RGBD camera: Unified benchmark and baselines," Proc. IEEE Int'l Conf. Computer Vision, 2013.

[38] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al., "Imagenet large scale visual recognition challenge," IJCV, vol. 115, no. 3, pp. 211–252, 2015.

[39] Jonathan Long, Evan Shelhamer, and Trevor Darrell, "Fully convolutional networks for semantic segmentation," in CVPR, 2015, pp. 3431–3440.

[40] ArnoldWMSmeulders, Dung MChu, Rita Cucchiara, Simone Calderara, Afshin Dehghan, and Mubarak Shah, "Visual tracking: An experimental survey," PAMI, vol. 36, no. 7, pp. 1442–1468, 2014.

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

[42] Martin Danelljan, Gustav H¨ager, Fahad Khan, and Michael Felsberg, "Accurate scale

estimation for robust visual tracking," in British Machine Vision Conference, Nottingham, September 1-5, 2014. BMVA Press, 2014.

[43] K. Fukunaga, Introduction to Statistical Pattern Recognition (2Nd Ed.). San Diego, CA, USA: Academic Press Professional, Inc., 1990.

[44] L. R. Rabiner and B. H. Juang, "An introduction to hidden markov models," IEEE ASSp Magazine, 1986.

[45] C. Liu and H. Wechsler, "Gabor feature based classification using the enhanced fisher linear discriminant model for face recognition," IEEE Transactions on Image Processing, vol. 11, no. 4, pp. 467– 476, Apr 2002.

[46] M. Riesenhuber and T. Poggio, "Hierarchical models of object recognition in cortex," Nature Neuroscience, vol. 2, pp. 1019–1025, 1999.

[47] T. Ahonen, A. Hadid, and M. Pietikainen, "Face description with local binary patterns: Application to face recognition," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 28, no. 12, pp. 2037–2041, Dec 2006.

[48] B. Moghaddam and M.-H. Yang, "Learning gender with support faces," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24, no. 5, pp. 707–711, May 2002.

[49] I. Ullah, M. Hussain, G. Muhammad, H. Aboalsamh, G. Bebis, and A. M. Mirza, "Gender recognition from face images with local wld descriptor," in 2012 19th

International Conference on Systems, Signals and Image Processing (IWSSIP), April 2012, pp. 417–420.

[50] J. Chen, S. Shan, C. He, G. Zhao, M. Pietikainen, X. Chen, and W. Gao, “Wld: A robust local image descriptor,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 32, no. 9, pp. 1705–1720, Sept 2010.

[51] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in Advances in Neural Information Processing Systems 25, 2012, pp. 1097–1105.

[52] G. Levi and T. Hassncer, “Age and gender classification using convolutional neural networks,” in 2015 IEEE Conference o [13]nComputer Vision and Pattern Recognition Workshops (CVPRW), June 2015, pp. 34–42.

[53] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” in ICLR, 2015.

[54] 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,” International Journal of Computer Vision (IJCV), vol. 115, no. 3, pp. 211–252, 2015.

[55] X. T. Yuan and S. Yan, “Visual classification with multi-task joint sparse representation,” in Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on, June 2010, pp. 3493–3500.

[56] T. Zhang, B. Ghanem, S. Liu, and N. Ahuja, “Robust visual tracking via structured multi-task sparse learning,” International Journal of Computer Vision, vol. 101, no. 2, pp. 367–383, 2012.

[57] Z. Zhang, P. Luo, C. C. Loy, and X. Tang, Computer Vision – ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part VI. Cham: Springer International Publishing, 2014, ch. Facial Landmark Detection by Deep Multitask Learning, pp. 94–108.

[58] Y. Sun, X. Wang, and X. Tang, “Deep convolutional network cascade for facial point detection,” in Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on, 2013, pp. 3476–3483.

[59] Y. H. Kwon and N. da Vitoria Lobo, “Age classification from facial images,” in Computer Vision and Pattern Recognition, 1994. Proceedings CVPR ’94., 1994 IEEE Computer Society Conference on, Jun 1994, pp. 762–767.

[60] N. Ramanathan and R. Chellappa, “Modeling age progression in young faces,” in 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’06), vol. 1, June 2006, pp. 387–394.

[61] T. F. Cootes, G. J. Edwards, and C. J. Taylor, “Active appearance models,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 23, no. 6, pp. 681–685, Jun 2001.

[62] A. Lanitis, C. Draganova, and C. Christodoulou, “Comparing different classifiers for

automatic age estimation," IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), vol. 34, no. 1, pp. 621–628, 2004.

[63] A. Lanitis, C. J. Taylor, and T. F. Cootes, "Toward automatic simulation of aging effects on face images," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24, no. 4, pp. 442–455, Apr 2002.

[64] X. Geng, Z. H. Zhou, and K. Smith-Miles, "Automatic age estimation based on facial aging patterns," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 29, no. 12, pp. 2234–2240, Dec 2007.

[65] Y. Fu, Y. Xu, and T. S. Huang, "Estimating human age by manifold analysis of face pictures and regression on aging features," in 2007 IEEE International Conference on Multimedia and Expo, July 2007, pp. 1383–1386.

[66] J. Hayashi, M. Yasumoto, H. Ito, and H. Koshimizu, "Method for estimating and modeling age and gender using facial image processing," in Virtual Systems and Multimedia, 2001. Proceedings. Seventh International Conference on, 2001, pp. 439–448.

[67] S. Yan, M. Liu, and T. S. Huang, "Extracting age information from local spatially flexible patches," in 2008 IEEE International Conference on Acoustics, Speech and Signal Processing, March 2008, pp. 737–740.

[68] Y. H. Kwon and N. da Vitoria Lobo, "Age classification from facial images," in Computer Vision and Pattern Recognition, 1994. Proceedings CVPR '94., 1994 IEEE

Computer Society Conference on, Jun 1994, pp. 762–767.

[69] G. Guo, Y. Fu, C. R. Dyer, and T. S. Huang, “Image-based human age estimation by manifold learning and locally adjusted robust regression,” IEEE Transactions on Image Processing, vol. 17, no. 7, pp. 1178–1188, July 2008.

[70] G. W. Cottrell and J. Metcalfe, “Empath: Face, emotion, and gender recognition using holons,” in Proceedings of the 1990 Conference on Advances in Neural Information Processing Systems 3, ser. NIPS-3, 1990, pp. 564–571.

[71] B. Poggio, R. Brunelli, and T. Poggio, “Hyberbf networks for gender classification,” in Proc. DARPA Image Understanding Workshop, 1995.

[72] L. Wiskott, J. M. Fellous, N. Kruger, and C. von der Malsburg,“Face recognition by elastic bunch graph matching,” in Image Processing, 1997. Proceedings., International Conference on, vol. 1, Oct 1997, pp. 129–132 vol.1.

[73] M. J. Lyons, J. Budynek, A. Plante, and S. Akamatsu, “Classifying facial attributes using a 2-d gabor wavelet representation and discriminant analysis,” in Automatic Face and Gesture Recognition, 2000. Proceedings. Fourth IEEE International Conference on, 2000, pp. 202–207.

[74] Z. Sun, G. Bebis, X. Yuan, and S. J. Louis, “Genetic feature subset selection for gender classification: a comparison study,” in Applications of Computer Vision, 2002. (WACV 2002). Proceedings. Sixth IEEE Workshop on, 2002, pp. 165–170.

[75] A. Jain and J. Huang, “Integrating independent components and linear discriminant analysis for gender classification,” in Automatic Face and Gesture Recognition, 2004. Proceedings. Sixth IEEE International Conference on, May 2004, pp. 159–163.

[76] N. P. Costen, M. Brown, and S. Akamatsu, “Sparse models for gender classification,” in Automatic Face and Gesture Recognition, 2004. Proceedings. Sixth IEEE International Conference on, May 2004, pp. 201–206.

[77] B. Moghaddam and M.-H. Yang, “Learning gender with support faces,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24, no. 5, pp. 707–711, May 2002.

[78] Y. LeCun, B. E. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. E. Hubbard, and L. D. Jackel, “Handwritten digit recognition with a back-propagation network,” in Advances in Neural Information Processing Systems 2, D. S. Touretzky, Ed., 1990, pp. 396–404.

[79] E. Eidinger, R. Enbar, and T. Hassner, “Age and gender estimation of unfiltered faces,” IEEE Transactions on Information Forensics and Security, vol. 9, no. 12, pp. 2170–2179, Dec 2014.

[80] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun, “Deep Residual Learning for Image Recognition,” arXiv preprint arXiv:1512.03385, 2015.