Advancing a Machine's Visual Awareness of People

Thesis by
David C. Hall

In Partial Fulfillment of the Requirements for the
degree of
Doctor of Philosophy

Caltech

CALIFORNIA INSTITUTE OF TECHNOLOGY
Pasadena, California

2017
Defended May 25, 2017

ACKNOWLEDGEMENTS

I would like to thank my adviser Pietro Perona for not only his guidance but for giving me the opportunity to follow my interests and study independently.

To my friends and colleagues in the Vision Lab: Grant Van Horn, Matteo Ruggero Ronchi, Oisin Mac Aodha, Ron Appel, Alvita Tran, Mason McGill, Joe Marino, Eyrun Eyjolfsdottir, Serim Ryou, Cristina Segalin, Steve Branson and Michael Maire, thank you for sharing your thoughts and ideas and of course your friendship.

To Katie for putting up with me and to the rest of my family for being understanding of me being so far away.

ABSTRACT

Methods to advance a machine's visual awareness of people with a focus on understanding 'who is where' in video are presented. 'Who' is used in a broad sense that includes not only the identity of a person but attributes of that person as well. Efforts are focused on improving algorithms in four areas of visual recognition: detection, tracking, fine-grained classification and person reidentification.

Each of these problems appear to be quite different on the surface; however, there are two broader questions that are answered across each of the works. The first, the machine is able to make better predictions when it has access to the extra information that is available in video. The second, that it is possible to learn on-the-fly from single examples. How each work contributes to answering these over-arching questions as well as its specific contributions to the relevant problem domain are as follows:

The first problem studied is one-shot, real-time, instance detection. Given a single image of a person, the task for the machine is to learn a detector that is specific to that individual rather than to an entire category such as faces or pedestrians. In subsequent images, the individual detector indicates the size and location of that particular person in the image. The learning must be done in real-time. To solve this problem, the proposed method starts with a pre-trained boosted category detector from which an individual-object detector is trained, with near-zero computational cost, through elementary manipulations of the thresholds of the category detector. Experiments on two challenging pedestrian and face datasets indicate that it is indeed possible to learn identity classifiers in real-time; besides being faster-trained, the proposed classifier has better detection rates than previous methods.

The second problem studied is real-time tracking. Given the initial location of a target person, the task for the machine is to determine the size and location of the target person in subsequent video frames, in real-time. The method proposed for solving this problem treats tracking as a repeated detection problem where potential targets are identified with a pre-trained boosted person detector and identity across frames is established by individual-specific detectors. The individual-specific detectors are learnt using the method proposed to solve the first problem. The proposed algorithm runs in real-time and is robust to drift. The tracking algorithm is benchmarked against nine state-of-the-art trackers on two benchmark datasets. Results show that the proposed method is 10% more accurate and nearly as fast as the fastest of the competing algorithms, and it is as accurate but 20 times faster than the most accurate of the competing algorithms.

The third problem studied is the fine-grained classification of people. Given an image of a person, the task for the machine is to estimate characteristics of that person such as age, clothing style, sex, occupation, social status, ethnicity, emotional state and/or body type. Since fine-grained classification using the entire human body is a relatively unexplored area, a large video dataset was collected. To solve this problem, a method that uses deep neural networks and video of a person is proposed. Results show that the class average accuracy when combining information from a sequence of images of an individual and then predicting the label is 3.5-7.1% better than independently predicting the label of each image, when severely under-represented classes are ignored.

The final problem studied is person reidentification. Given an image of a person, the task for the machine is to find images that match the identity of that person from a large set of candidate images. This is a challenging task since images of the same individual can vary significantly due to changes in clothing, viewpoint, pose, lighting and background. The method proposed for solving this problem is a two-stage deep neural network architecture that uses body part patches as inputs rather than an entire image of a person. Experiments show that rank-1 matching rates increase by 22-25.6% on benchmark datasets when compared to state-of-the-art methods.

PUBLISHED CONTENT AND CONTRIBUTIONS

[1] D. Hall and P. Perona. “From Categories to Individuals in Real Time — A Unified Boosting Approach”. In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2014. DOI: 10.1109/cvpr.2014.30.
D.H participated in designing the project, developing the method, running the experiments and writing the manuscript.

[2] D. Hall and P. Perona. “Online, Real-Time Tracking Using a Category-to-Individual Detector”. In: *European Conference on Computer Vision (ECCV)*. 2014. DOI: 10.1007/978-3-319-10590-1_24.
D.H participated in designing the project, developing the method, running the experiments and writing the manuscript.

[3] D. Hall and P. Perona. “Fine-Grained Classification of Pedestrians in Video: Benchmark and State of the Art”. In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015. DOI: 10.1109/cvpr.2015.7299187.
D.H participated in designing the project, developing the method, running the experiments and writing the manuscript.

[4] D. Hall and P. Perona. “Putting Pose into Person Reidentification”. In: *In Submission to the International Conference on Computer Vision (ICCV)*. 2017.
D.H participated in designing the project, developing the method, running the experiments and writing the manuscript.

TABLE OF CONTENTS

Chapter 1

INTRODUCTION

Intelligent machines are everywhere. From a simple cruise control system in a vehicle, to a smart-phone that can answer questions from speech or identify bird species from images, to machines that defeat humans in complex games such as chess or go [9]. If we believe that people are the most important component of a machine's environment, then we will want to make machines capable of interacting with people. How then does one design an intelligent machine that has such a capability?

An intelligent machine has three basic components. 1) A set of **sensors** that receive information about the world; possible sensors include those that are responsive to visual, auditory, tactile, or olfactory inputs. 2) A **cognitive** component that interprets the input, maybe using past experiences or other information, to make a decision about an action that needs to be taken. This is the intelligent part of the machine. 3) A set of **effectors** that allow the machine to interact with the world based on the decided action. Possible effectors include those that create visual, auditory, tactile or olfactory outputs.

Machine vision concerns itself with designing the **cognitive** component of an intelligent machine that is responsible for understanding and interpreting **visual input**. While other sensory inputs are useful and important, the focus on visual input is because it is informative and fast, which is why it is used by humans so much—50% of our neural tissue is directly or indirectly related to vision. For the rest of this work, the discussion will focus on the methods used by machines to understand visual information.

To design algorithms that are responsible for **visual cognition** it is useful to think through a scenario that allows for a rich and natural interaction between humans and machines. Consider the case of a robot monitoring an operating room, and its job is to keep an eye out for mistakes (it might warn the surgeon if an implement has been left inside the patient). The robot would thus need to understand the following things about people from the stream of images it is receiving[1]:

[1] Although we discuss people specifically, the same thought process can be applied to other objects since the robot also needs to have an understanding about the other things in the room.

1. **Where are all the people in space and time.** The robot needs to know the location of all the people that are within its visual field. In machine vision, the task of finding an object in an image is called detection [5, 12, 4]. If the objects are to be followed in video, the task is called tracking [13, 11]. A finer level of detection is determining the location of the parts of an object, such as the position of the hands, fingers or eyes of a person. This is called pose estimation [8, 10], which is essentially a detection task with a set of constraints about where the parts can be relative to each other.

2. **What are the characteristics or attributes of these people.** Once the robot knows where people are, it can then start to make higher level judgements about the characteristics of these people, determining their age, clothing style, occupation, ethnicity and/or identity. This task is known as fine-grained classification [3, 14] and allows the robot to distinguish between the people that are doctors from those that are nurses. Determining identity is the 'finest' possible fine-grained classification and is treated separately in machine vision due to its importance. Assigning unique identities to individuals allows the robot to keep track of people, particularly when they exit and re-enter the robot's visual field. This task is known as person reidentification [2].

3. **What are these people doing.** Once the robot knows 'who is where' (who is used in a broad sense that includes not only the identity of a person but attributes of that person as well) it can make judgements about what fine-grained tasks people are doing such as injecting, reaching or cutting. In machine vision this is called action recognition [6]. By combining a sequence of actions the robot can then make a judgement about the activity that is occurring, such as the patient is being anaesthetised. This is called activity recognition [1, 7].

4. **What should I be doing.** Once the robot has an understanding of what is happening around it, it can now decide how to act based on this information and its own goals. Knowing which people are where and what they are doing helps the robot determine if they need to warn a specific individual because they are doing something hazardous or have left an object behind.

By thinking through what a machine needs to perceive from a sequence of images, it is obvious that visual cognition for machines is a diverse area of research with elements that build upon each other. Since the more complex, higher level tasks of planning and activity recognition depend on the accuracy of the lower level tasks, I focus my efforts on improving the algorithms for detection, tracking, fine-grained classification and person reidentification.

Each of these problems appear to be quite different on the surface; however, there are two broader questions that are explored across each of the works. The first, how do you do visual recognition in video? The machine may be able to make better predictions when it has access to the extra information that is available in video. The second, how do you learn on-the-fly from single examples? Updating the machine's understanding of the world quickly and with small amounts of data gives the machine the ability to interact with humans in real-time. To summarise:

This thesis seeks to advance the methods for machines to be visually aware of people with a focus on understanding who, along with their attributes, is where in video.

The reader will find a collection of self-contained chapters. Each chapter advances the methods for one of the low level tasks of detection, tracking, fine-grained classification and person reidentification but also addresses the broader problems of visual recognition in video and learning on-the-fly from single examples. Chapters 2-4 have been adapted from peer-reviewed publications. Chapter 5 is in submission at the time of writing. Related work is discussed separately in each chapter. An outline of the thesis follows.

Chapter 2 addresses the problem of instance detection. Instance detection involves learning detectors for specific individuals rather than an entire category such as faces or pedestrians. In this chapter a method for real-time, online training of individual detectors from individuals that are detected by a category detector is presented. Two questions are answered, assuming that a category detector is available. The first is how should individual detectors be designed so that their *additional* run-time cost is small or zero; the second is how can such individual detectors be trained on-the-fly with *minimal computational cost* once one or more training examples become available from the category detector. A unified boosting-based approach for simultaneous category and individual detection is proposed.

Chapter 3 addresses the tracking problem. In this chapter a method for online, real-time tracking of people is presented. Tracking is treated as a repeated detection problem where potential target objects are identified with a pre-trained category detector and object identity across frames is established by individual-specific detectors. The individual-specific detectors are learnt using the method proposed in Chapter 2. The proposed tracking algorithm runs in real-time and is robust to drift.

Chapter 4 addresses the fine-grained classification problem. In this chapter a public video dataset—Caltech Roadside Pedestrians (CRP)—is collected to further advance the state-of-the-art in fine-grained classification of people using the entire human body. A unified model for the fine-grained classification of people using single images is proposed. A study of how temporal information can be utilised for the fine-grained classification of people is also conducted.

Chapter 5 addresses the person reidentification problem. In this chapter, a method for reidentifying people using pose information is presented. A two-stage deep neural network architecture that uses body part patches as inputs rather than an entire image of a person is proposed. Different strategies for combining the per part information are explored.

Finally, in Chapter 6, findings are summarised and a discussion of future work is presented.

References

[1] J. Aggarwal and M. Ryoo. "Human Activity Analysis: A Review". In: *ACM Comput. Surv.* 43.3 (2011), 16:1–16:43. DOI: 10.1145/1922649.1922653.

[2] E. Ahmed, M. Jones, and T. K. Marks. "An Improved Deep Learning Architecture for Person Re-Identification". In: *CVPR*. 2015. DOI: 10.1109/CVPR.2015.7299016.

[3] S. Branson et al. "Improved Bird Species Recognition Using Pose Normalized Deep Convolutional Nets". In: *BMVC*. 2014. DOI: 10.5244/C.28.87.

[4] J. Deng et al. "ImageNet: A Large-Scale Hierarchical Image Database". In: *CVPR*. 2009. DOI: 10.1109/CVPR.2009.5206848.

[5] P. Dollár et al. "Pedestrian Detection: An Evaluation of the State of the Art". In: *PAMI* 34.4 (Apr. 2012), pp. 743–61. DOI: 10.1109/TPAMI.2011.155.

[6] S. Herath, M. Harandi, and F. Porikli. "Going Deeper into Action Recognition: A Survey". In: *Image and Vision Computing* 60 (2017), pp. 4–21. DOI: 10.1016/j.imavis.2017.01.010.

[7] O. D. Lara and M. A. Labrador. "A Survey on Human Activity Recognition using Wearable Sensors". In: *IEEE Communications Surveys Tutorials* 15.3 (2013), pp. 1192–1209. DOI: 10.1109/SURV.2012.110112.00192.

[8] A. Newell, K. Yang, and J. Deng. "Stacked Hourglass Networks for Human Pose Estimation". In: *ECCV*. 2016. DOI: 10.1007/978-3-319-46484-8.

[9] D. Silver et al. “Mastering the Game of Go with Deep Neural Networks and Tree Search”. In: *Nature* 529.7587 (2016), pp. 484–489. doi: 10.1038/nature16961.

[10] S. E. Wei et al. “Convolutional Pose Machines”. In: *CVPR*. 2016. doi: 10.1109/CVPR.2016.511.

[11] Y. Wu, J. Lim, and M.-H. Yang. “Online Object Tracking: A Benchmark”. In: *CVPR*. 2013. doi: 10.1109/CVPR.2013.312.

[12] M.-H. Yang, D. J. Kriegman, and N. Ahuja. “Detecting Faces in Images: A Survey”. In: *PAMI* 24.1 (2002), pp. 34–58. doi: 10.1109/34.982883.

[13] A. Yilmaz, O. Javed, and M. Shah. “Object tracking”. In: *ACM Computing Surveys* 38.4 (Dec. 2006), 13–es. doi: 10.1145/1177352.1177355.

[14] B. Zhao et al. “A Survey on Deep Learning-based Fine-grained Object Classification and Semantic Segmentation”. In: *International Journal of Automation and Computing* 14.2 (2017), pp. 119–135. doi: 10.1007/s11633-017-1053-3.

Chapter 2

INSTANCE DETECTION

The contents of this chapter are from the peer-reviewed publication "From Categories to Individuals in Real Time — A Unified Boosting Approach" by D. Hall and P. Perona, appearing at CVPR 2014[1].

2.1 Abstract

A method for online, real-time learning of individual-object detectors is presented. Starting with a pre-trained boosted category detector, an individual-object detector is trained with near-zero computational cost. The individual detector is obtained by using the same feature cascade as the category detector along with elementary manipulations of the thresholds of the weak classifiers. This is ideal for online operation on a video stream or for interactive learning. Applications addressed by this technique are reidentification and individual tracking. Experiments on four challenging pedestrian and face datasets indicate that it is indeed possible to learn identity classifiers in real-time; besides being faster-trained, our classifier has better detection rates than previous methods on two of the datasets.

2.2 Introduction

Detecting objects in image collections and video is a rich area of application of visual recognition. Technical approaches change significantly depending on whether one focuses on *individual-objects* [36] or on *categories* [6, 16, 19] and the two challenges are pursued as distinct research questions. While this separation is useful in academic research, real-world systems require a combination of category and individual detection.

For concreteness, we describe two such scenarios. The first is *tracking*. Applications include tracking of pedestrians in railway stations and airports, vehicles on the road for traffic monitoring and faces for interfacing people with computers. A common approach used is tracking-by-repeated-detection. In its simplest form this consists of a frame-by-frame category detector followed by an algorithm that combines detections across space and time into trajectories [3, 5]. Trajectory smoothness constraints confer a degree of robustness to false detections; the

[1] Project Website: http://vision.caltech.edu/~dhall/projects/CategoriesToIndividuals

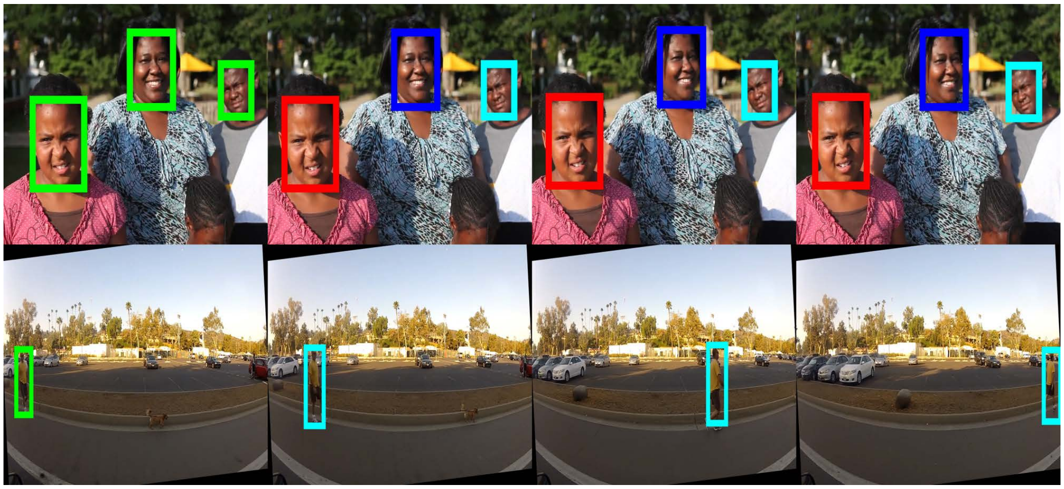


Figure 2.1: Tracking individuals across a video sequence: faces (top) and pedestrians (bottom). The first column shows the detections made by a category detector. These detections are used to initialise three individual face detectors, and a single individual pedestrian detector, and are then evaluated on the subsequent frames in the video sequence. Individual detector outputs are colour-coded.

main challenge is continuing trajectories when detection fails over multiple frames because of occlusion, unusual pose or unfavourable lighting conditions. Once an object (a pedestrian) has been detected by the category detector (or by a human operator), tracking is made more robust by training an individual-object detector, exploiting the specifics of the individual's appearance (a person with a red sweater and backpack) [39].

Reidentification is another scenario with applications in video surveillance [10] and content-based indexing of image collections, consumer videos and commercial video libraries. After a category detector detects instances of the category, individual detectors are trained to cluster and classify the individuals that appear in the collection [17]. Individual reidentification across networks of cameras is similarly important [7, 8, 28, 46].

It is clear from these examples that it is useful to detect objects both as members of categories and as individuals. In the first scenario, individual detectors trained on-the-fly improve tracking robustness. In the second scenario, individual classifiers reveal recurring individuals in an entire collection or video stream. In both cases it is crucial that an individual detector is trained in real-time and that its runtime cost does not add significantly to the overall computational cost of the system.

Here we present a method for real-time, online training of individual detectors from individuals that are detected by a category detector. We make three main contributions:

1. A unified boosting-based approach for simultaneous category and individual detection.

2. A method for training individual detectors in real-time from a single training example.

3. Two novel challenging datasets of faces and pedestrians.

2.3 Related work

Researchers in visual categorisation agree that objects are best represented as constellations of visually distinctive parts that appear in flexible geometrical arrangements [21, 34, 6, 19]. A variety of practical approaches to detecting parts and representing mutual positions have been proposed, where the representation of shape is either explicit [20, 19] or implicit [35, 40, 44, 14]; best performance is currently obtained with discriminatively trained part detectors [14, 19]. This work is based on boosted cascades of classifiers [44, 4, 14] because they deliver state-of-the-art detection performance at video-rate computational speeds [2].

Researchers focusing on individual detection [36] and re-identification [10] focus both on the design of (domain-specific) features [28, 7, 46] and on efficient algorithms for detection and classification [36]. In our work we are feature-agnostic, in that our framework allows the implementation of a large variety of different features, and we rely on the computational efficiency of cascaded boosted classifiers.

Online learning of detectors for tracking individual objects, given an operator-supplied initial training window, is a topic of much interest [9, 31]; the main challenge is *drifting* from the original target. The closest work to our own are the online boosted trackers of Grabner *et al*. [23, 24, 25]. In their work, boosted individual-object detectors are trained online and are paired with a *prior* to limit drift. The individual detectors operate at frame rates of between 10–15 frames-per-second on a video with a resolution of 640x480; however, this cost is in addition to the cost of running a category detector, the output of which initialises the individual detector.

Our work aims to produce a unified approach for simultaneous category and individual detection to ensure that real-time operation can be achieved. We focus on two questions that have, to our knowledge, not yet been studied: assuming that a category detector is available, (a) how to design individual detectors whose *ad-*

ditional run-time cost is small or zero; (b) how to train such individual detectors on-the-fly with *minimal computational cost* once one or more training examples become available from the category detector.

2.4 Approach

Our approach is based on using *cascaded boosted classifiers* both for category and individual detection [43, 44, 14, 13]. Detectors of this form have been shown to be fast and have state-of-the-art detection performance [2]. In order to make this paper self-contained, we review cascaded boosted classifiers (Sec. 2.4), discuss the implementation of category detectors (Sec. 2.4) and finally outline the approach for designing individual detectors (Sec. 2.4).

Boosting

A boosted classifier takes feature vector $\mathbf{x} \in \mathbb{R}^D$ as input and outputs a binary decision:

$$H(\mathbf{x}) = \operatorname{sign}\left(\sum_{m=1}^{M} \alpha_m h_m(\mathbf{x}) - \tau\right) \tag{2.1}$$

where the threshold τ is chosen to produce the desired tradeoff between false reject rate and false alarm rate. Given a labelled training set $\{\mathbf{x}_i, y_i\}_i$ of N samples, the boosted classifier is trained by greedily minimising a loss function (which depends on the type of boosting being used: AdaBoost, LogitBoost, *etc.*). This means that at each iteration m up until the maximum number of iterations M an optimal weak classifier $h_m(\mathbf{x})$ and weight α_m are selected. For training, each data sample x_i is assigned a weight w_i^m (depends on the loss function). At each iteration, samples that are classified incorrectly are weighted more heavily, which means the penalty for classifying them incorrectly in subsequent iterations increases.

Category Detector

In this work category detectors are trained offline using AdaBoost [22]. The family of weak classifiers used are stumps. This means that given an input $\mathbf{x} \in \mathbb{R}^D$, the decision only depends on the j-th dimension of $\mathbf{x}$, a threshold $\theta \in \mathbb{R}$ and a polarity $p \in \{\pm 1\}$

$$h_m(\mathbf{x}) = \begin{cases} 1, & p_m x_{j_m} > p_m \theta_m \\ -1, & \text{otherwise} \end{cases} \tag{2.2}$$

During training, the optimal weak classifier at the m-th iteration of boosting is selected by choosing j, θ and p so that the number of the N weighted training examples that are misclassified is minimised. Choosing these parameters at each of the M iterations is $O(MND)$ and is the most computationally expensive part of training a boosted classifier. Note that any boosting method and decision trees of any depth could be used to train the category detector; our proposed method is agnostic to these choices. For the sake of clarity, in the following discussion we will continue to refer to AdaBoost and decision stumps.

Individual Detectors

The proposed approach for designing individual detectors relies on four key principles: 1) the individual detector has the form of a cascade of boosted classifiers (Eqn. 2.1); 2) an individual detector is learnt from a single instance of the individual; 3) the *training* and 4) the *runtime* costs of an individual detector must be minimal to guarantee online, real-time operation.

The most obvious strategy for training an individual detector is to repeat the AdaBoost training process using an object identified by the category detector as a positive training example. Recent work has made the training stage of AdaBoost faster [1]; however, it is still a computationally expensive process and remains ill-suited for real-time operation. In the following exposition we will look at the limitations of a traditional boosting approach and examine a set of constraints that can be placed on the individual detectors to avoid the computationally costly steps of a traditional boosted detector.

The first design goal requires individual detectors to be of the same form as the category detector. This is a reasonable restriction to place on the individual detectors since cascades are fast, making them suitable for real-time operation and their performance is state-of-the-art as has been previously mentioned. This requirement also ensures there is simplicity in design and a unified approach for both category and individual detection.

The second principle, that an individual detector is learnt from a single instance of an individual is also a reasonable requirement. Training using a traditional boosting approach is possible by jittering or transforming the original example. This results in multiple, slightly altered versions of the original instance which can all be used as positive training examples. The drawback here is that negative examples are now required. This either requires precomputed negatives to be stored in memory (which may be limited) or for negative examples to be mined online, which is another costly computation.

To ensure design goals 3) and 4) are satisfied it is important to examine the most computationally expensive steps in the object detection pipeline, the first of which is feature computation.

Computing features is expensive; however, some features have already been computed for the category detector. If we can re-use the same features for the individual detector, the additional runtime and training costs for an individual detector due to feature computation are zero. For this reason, **individual detectors will be constrained to only use features that have already been computed for category detection**.

A second computationally expensive stage to consider is feature selection during training. This is equivalent to choosing parameter j at the m-th boosting iteration (Sec. 2.4). Typically, D, the dimensionality of the feature space is large so performing this optimisation is costly; however, this optimisation can be avoided if the **individual detectors are constrained to use only the M features that were selected by AdaBoost for the category detector**. We will denote this set of features by $\mathbf{J} = (j_1, \dots, j_M)$ where $j_m \in \{1, \dots, D\}$ and the importance of each feature through the weights $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_M)$. The additional training cost due to feature selection is thus zero.

It is not intuitive that category detection features (features that are good at distinguishing faces from background), are also useful for individual identification (distinguishing faces from other faces). It is reasonable to expect that the features AdaBoost would select for a face detector are features that are *common* to all faces; consequently, these common features should be uninformative for distinguishing *between* faces. Consequently, constraining the individual detectors to only use the M features of the category detector ought to doom it to failure. However, this intuition is not necessarily correct.

For a category detector to perform well, it needs to be able to detect many different types of faces in different lighting conditions. It is not necessary for an individual stump to cover the complete range of feature values; it only needs to capture a narrow range. Breadth is achieved by combining multiple stumps. It would then be reasonable to expect that the feature distributions for an individual are localised within narrower intervals that are contained within the category distribution for that particular feature.

Plots in Figure 2.2 show the empirical distribution of nine features from faces used

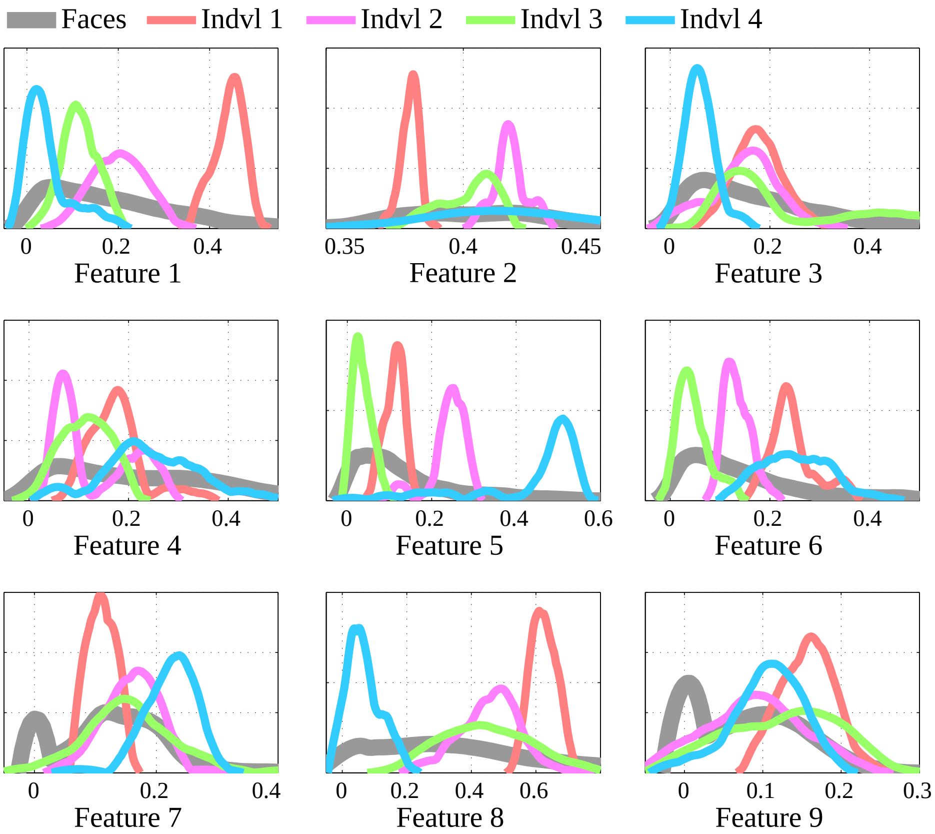


Figure 2.2: The empirical distributions of a set of faces and four individual faces across the first nine features selected by AdaBoost for the category detector. The set of faces was used to train the category detector (refer to Sec. 2.6 for details). The four individuals represent a single sequence of an individual lasting for at least fifty-five frames from a test video in the FPOQ dataset (Sec. 2.6).

to train the category detector (grey) as well as the distributions for four different individuals (colour); the individual distributions are obtained from video sequences lasting at least fifty-five frames from a test video in the FPOQ dataset (Sec. 2.6). (In video, a sequence is a consecutive set of frames in which an individual appears. Individuals occur in multiple sequences across the length of the video.) Each subplot is for one of the first nine features selected by AdaBoost for the category detector. This plot suggests that the distribution of a feature for a particular individual is localised within the broader category distribution. To substantiate this claim, a statistical analysis across many different individuals is required.

Let $X^+ \in \mathbb{R}^{N\text{x}M}$ be the matrix of M-dimensional feature vectors of the N positive examples of a category. The N positive examples consist of multiple instances of the same individual, possibly under different pose and lighting conditions. The range

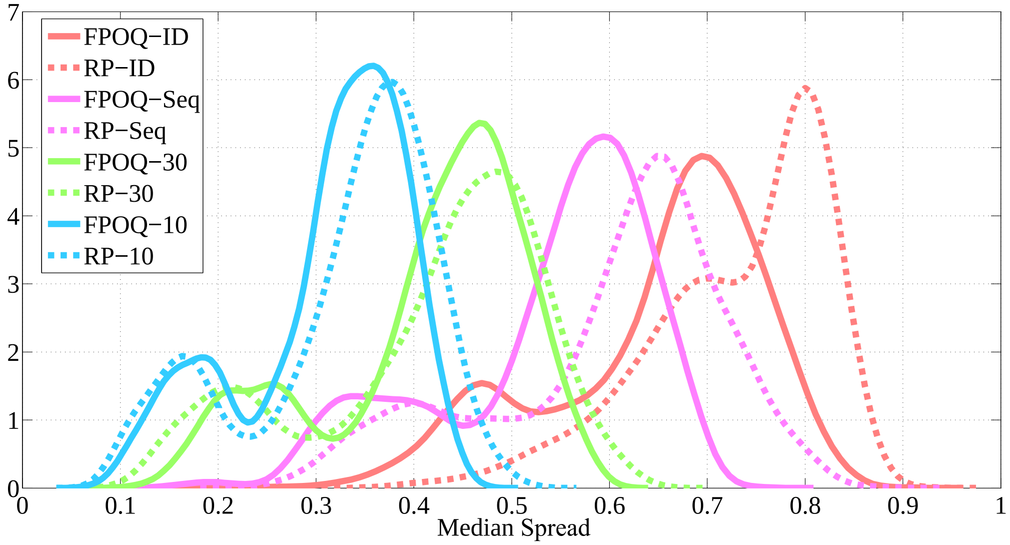


Figure 2.3: The empirical distributions of the median spread (Eqn. 2.3) for the M features selected by a boosted category detector of the faces in the FPOQ (solid) and of the pedestrians in the CRP (dashed) datasets (Sec. 2.6). The red curves correspond to the median spread of individuals across multiple sequences, the pink to individuals in a single sequence, the green to 30 consecutive frames of an individual and the blue to 10 consecutive frames of an individual.

of feature j for the *entire category* can then be defined as $r_j^+ = Q_{0.95}(X_j) - Q_{0.05}(X_j)$ where Q_p is the p-th quantile and X_j is the j-th column of X^+. The range of feature j for an *individual across multiple sequences* is $r_j^i = Q_{0.95}(X_j^i) - Q_{0.05}(X_j^i)$ where X_j^i is the j-th column of X^+ but only with the rows that correspond to individual i. The median spread l_j across all individuals for feature j is then defined as:

$$l_j = \underset{i}{\text{median}}\left(\frac{r_j^i}{r_j^+}\right) \tag{2.3}$$

The median spread of an individual in a single sequence, in 30 consecutive frames and in 10 consecutive frames is also considered and can be defined similarly. Figure 2.3 shows the empirical distributions of the median spread for the features for all faces in the FPOQ and all pedestrians in the CRP datasets (Sec. 2.6) .

Figure 2.3 suggests that reidentifying individuals within sequences (pink curves) or within 30 (green curves) or 10 (blue curves) frames of each other is possible since most features are localised (the spread is small with respect to the category distribution). It also suggests that reidentifying individuals across sequences (red curves) may be problematic since there are more features with higher spread values; however, the architecture of a cascaded boosted classifier provides some robustness to this variability between sequences. If there are enough features that exhibit limited variability (FPOQ (solid red) has a number of features with a spread of less than 0.5) then an individual detector may still classify an individual correctly across sequences because there is sufficient evidence to suggest that the individual is present.

The third and final computationally expensive stage in a traditional boosting approach is threshold selection during training. Even if features have already been selected (parameter j has been fixed), the optimal threshold θ, at the m-th boosting iteration still needs to be chosen (Sec. 2.4). This is once again computationally expensive; however, this optimisation can be avoided if we consider an alternative approach.

Selecting the thresholds for a single weak classifier $h'(x')$, which depends on a single feature $x' \in \mathbb{R}$, can be achieved at almost zero computational cost by using transfer learning. Consider the single instance of an individual that has been detected by the category detector and call γ' the value of feature x' for this instance. Figure 2.2 suggests that the distribution of features for an individual tends to be localised. The average spread σ' of feature x' across many individuals may be estimated offline using a validation set composed of images grouped by individual. An interval $(\gamma' - \beta\sigma', \gamma' + \beta\sigma')$ can then be defined where β is a free parameter that can be tuned experimentally. This interval represents the most likely values that the feature x' will take for the individual detected by the category detector. According to this strategy, the weak classifier $h'(x')$ may be obtained directly from one training example:

$$h'(x'; \gamma', \sigma') = \begin{cases} 1 & \gamma' - \beta\sigma' < x' < \gamma' + \beta\sigma' \\ -1 & \text{otherwise.} \end{cases} \tag{2.4}$$

This weak classifier provides evidence for an individual being present (absent) if the feature x' lies inside (outside) the interval $(\gamma' - \beta\sigma', \gamma' + \beta\sigma')$. The training cost for selecting the thresholds for a single weak classifier is thus a small constant.

Using the ideas presented, an individual detector in the form of a cascaded boosted classifier can now be constructed. Given the set of features $\mathbf{J}$ and weights $\boldsymbol{\alpha}$ that were selected for the category detector, the first instance $\mathbf{u}^k$ of the k-th individual detected by the category detector, and an estimate of the spread $\boldsymbol{\sigma} = (\sigma_1, \ldots, \sigma_M)$

of the features $\mathbf{J}$ a classifier $F^k(\mathbf{x})$ for the k-th individual can be defined by:

$$F^k(\mathbf{x}; \boldsymbol{\gamma}^k, \boldsymbol{\sigma}) = \sum_{m=1}^{M} \alpha_m h'(x_{j_m}; \gamma_m^k, \sigma_m) \tag{2.5}$$

where $\boldsymbol{\gamma}^k = (\gamma_1^k \dots \gamma_M^k)$ with $\gamma_m^k = u_{j_m}^k$. The total computational cost for learning an individual detector using the outlined approach is only $O(M)$. This is significantly less expensive than if the individual detector was trained using standard AdaBoost which is $O(DMN)$.

2.5 Datasets

To carry out the experiments it was necessary to collect two new challenging video datasets that contain many different individuals that reappear at different moments in time.

The first dataset is the Fifty People One Question (FPOQ) face dataset. It contains 6 videos with 222 annotated individuals across 725 sequences (in video, a sequence is a consecutive set of frames in which an individual appears). Each annotation contains the bounding box, the identity and the sequence number of the face. In total the are 68,676 bounding boxes, 78,181 frames and 57,274 frames that contain faces. The videos were collected from YouTube and involve either a single individual or groups of individuals being asked a question in front of a fixed camera. Their responses are edited in such a way so that an individual's response is interspersed between the responses of others. This means individuals can appear at any time point within the video. Examples of the different individuals as well as the different appearances of those individuals throughout the video are displayed in Figure 2.4. The face category detector was trained using 800 different faces extracted from single frames across 26 other videos (these videos are in the same style as the FPOQ videos). The faces used to train the face detector are not used during testing. The spread $\boldsymbol{\sigma}$ of the features selected by the face detector averaged over many individuals is estimated from a video in the FPOQ dataset. The other 5 videos are used for testing.

The second dataset is the Roadside Pedestrian (RP) dataset. It contains 2 videos with 170 annotated individuals across 263 sequences. In total there are 7450 bounding boxes, 77,450 frames and 5606 frames that contain pedestrians. Each video is captured by mounting a rightwards-pointing video camera to the roof of a car. The car then completes two laps of a ring road within a park where there are many walkers and joggers. This dataset is more challenging than the face dataset due to the

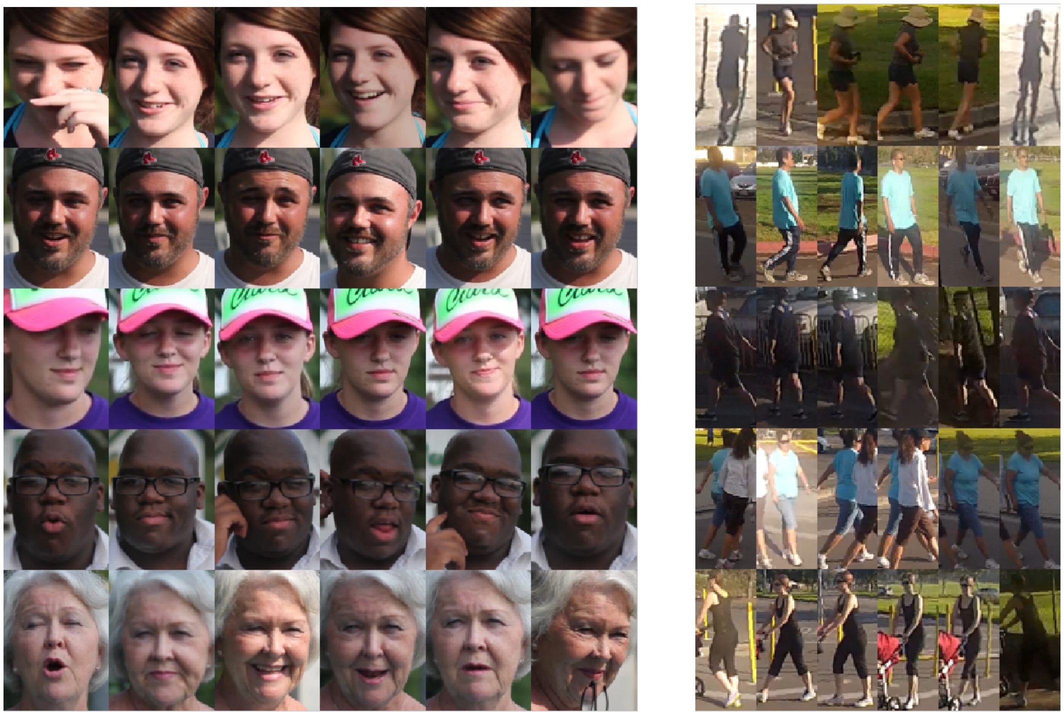


Figure 2.4: **(Left)** The faces of five different people from the FPOQ dataset. In digital versions of this work, clicking on the image will redirect you to YouTube to view one of the videos in the dataset. The faces in this dataset are quasi-frontal; lighting varies between overcast and direct sunlight; a variety of expressions are present as individuals are filmed while talking. **(Right)** Examples of five different pedestrians (one for each row) from the Roadside Pedestrian Dataset. The pedestrians show a wide range of poses, lighting conditions and backgrounds. The individuals were sampled randomly from a video in each of the datasets.

considerable differences in lighting and pose for an individual. Figure 2.4 displays a few examples of the pedestrians in this set. The pedestrian category detector was trained using 64 of the individuals in the RP dataset; σ was also estimated from this set. The remaining 106 individuals were used for testing.

For completeness we also evaluate our method on two existing re-identification datasets, VIPeR [26], which contains 632 person image pairs and ETHZ [42, 15] which contains multiple windows of people extracted from 3 video sequences. For conciseness we only report the results on ETHZ Sequence 2 which contains 35 persons across 1961 images.

2.6 Experiments

To assess the performance of our procedure for training individual detectors (we call it IDBoost), as well as to determine the limitations and applicability of the

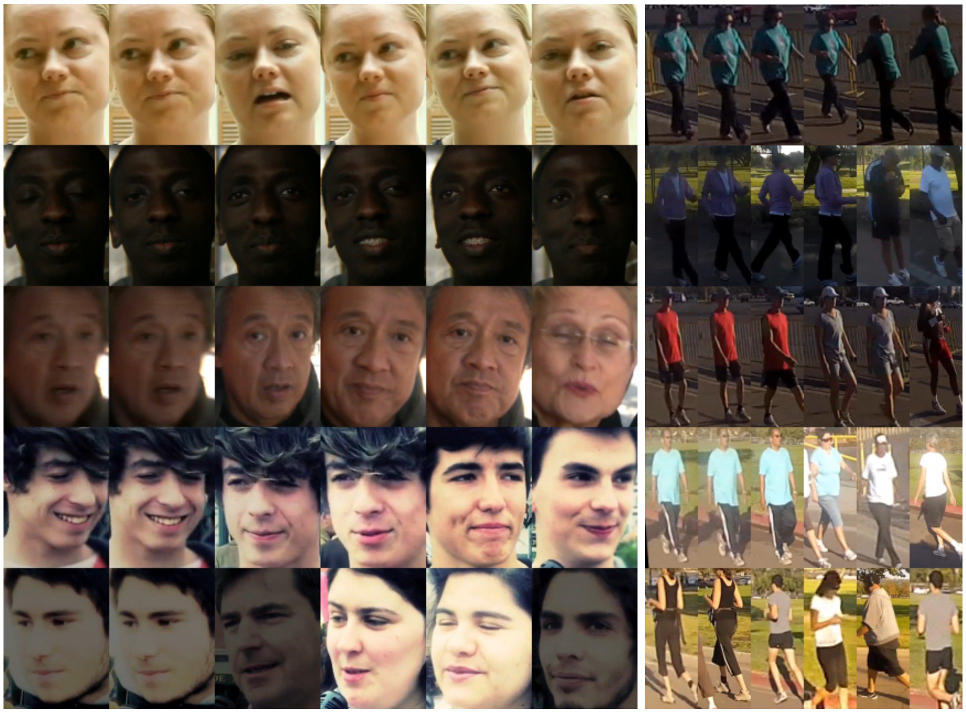


Figure 2.5: A demonstration of the reidentification of individuals in (left) the FPOQ and (right) the RP datasets using IDBoost. Column 1 shows the instance used to train the individual detector, it is the first instance of that individual detected by the category detector. For the FPOQ dataset, columns 2–6 show the top, 50th, 100th, 200th and 300th best scoring results returned by the individual detector after being evaluated on the ~10,000 detections made by the category detector. For the RP dataset columns 2–6 show the top, 5th, 10th, 20th and 30th best scoring results from ~2500 detections.

approach, two types of experiments, that illustrate two possible operating regimes for the individual detectors, are considered. The experiments are evaluated on two different categories: faces and pedestrians. All experiments are carried out using the multi-scale detection framework of Dollar *et al.* [13], with the channels of brightness, colour, gradient magnitude and gradient orientation used as features; the code is available in Dollar's publicly available Image and Video Matlab Toolbox[2]. In all experiments, the parameter $\beta = 0.6$; this choice ensured detectors operated at a fast enough rate whilst maintaining performance.

Reidentification

The first mode of operation for an individual detector is as a classifier. This means that the individual detector is evaluated on the single windows indicated by the

[2] https://github.com/pdollar/toolbox

category detector to contain an object of interest. Given a single example of an individual, the reidentification problem is a binary classification task where subsequent objects of interest are classified as either matching or not-matching that individual. This can be challenging since each instance of an individual varies in pose, lighting, background and occlusion. The reidentification experiments are run on both the FPOQ and the RP datasets.

The setup for this experiment is as follows:

1. The category detector extracts all instances of an object (faces in the FPOQ dataset and pedestrians in the RP dataset) in a video.
2. An individual detector is then trained for each of the individuals present in the video. Only the very first instance of an individual is ever used to train an individual detector. Each new individual is determined by a human operator.
3. Each individual detector is then applied to every other instance that was detected by the category detector in the video. A true positive occurs when an individual detector fires on the same individual that it was trained on.
4. An ROC curve is generated for each of the individual detectors and the average ROC across all individuals is then computed. The mean equal error rate is also reported.

Reidentification: Existing Methods

The performance of individual detectors trained using the proposed method (IDBoost) is compared to individual detectors trained using four other methods: AdaBoost [22]; the One-Shot Similarity score using LDA (OSS+LDA) [33] (code obtained from the authors website); KISSME [32], a metric learning algorithm that has state-of-the-art performance on the VIPeR dataset (code obtained from the authors website); and an l^2 distance detector as a baseline. These methods all use a single example of an individual for online training to provide a fair comparison. IDBoost and KISSME have an offline training component.

To learn an individual detector from a single example using AdaBoost, virtual positives are created by applying slight transformations to the example while negative examples are sampled from the background of the frame; this is similar to the initialisation stage of the online boosting trackers [23, 25].

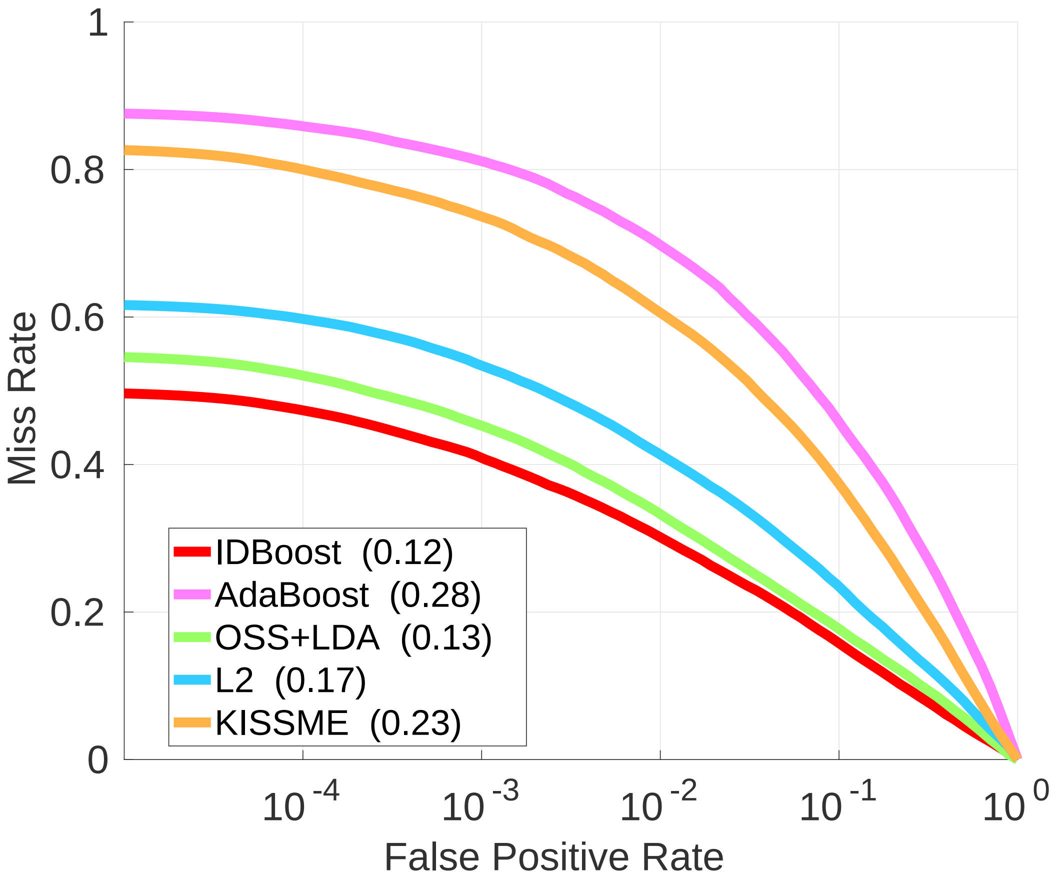


Figure 2.6: Reidentification performance. ROC curves for reidentification averaged over the 207 individuals in the FPOQ dataset. The mean equal error rate is also reported. IDBoost has the best mean equal error rate of 0.12. It is marginally better then OSS+LDA. In the FPOQ dataset there are 9375–15521 face detections per video

OSS+LDA returns a score that reflects the likelihood of an input vector belonging to the class defined by a single positive example and not to a class defined by a set of negative examples. The first instance of an individual is used as the single positive example and the set of first instance feature vectors of all the other individuals are used as the set of negative examples.

KISSME is an extremely fast, non-iterative method for learning a Mahalanobis metric. From a set of N exemplars the l^2 difference is computed for every pair of exemplars resulting in the distance matrix D. D is then partitioned into two matrices: D_0, which corresponds to pairs of exemplars that do not have the same identity and D_1, which corresponds to pairs of exemplars that do have the same identity. The metric is then formulated as a difference between the inverted covariance matrices of D_0 and D_1. In these experiments D_0 and D_1 are computed using the same validation

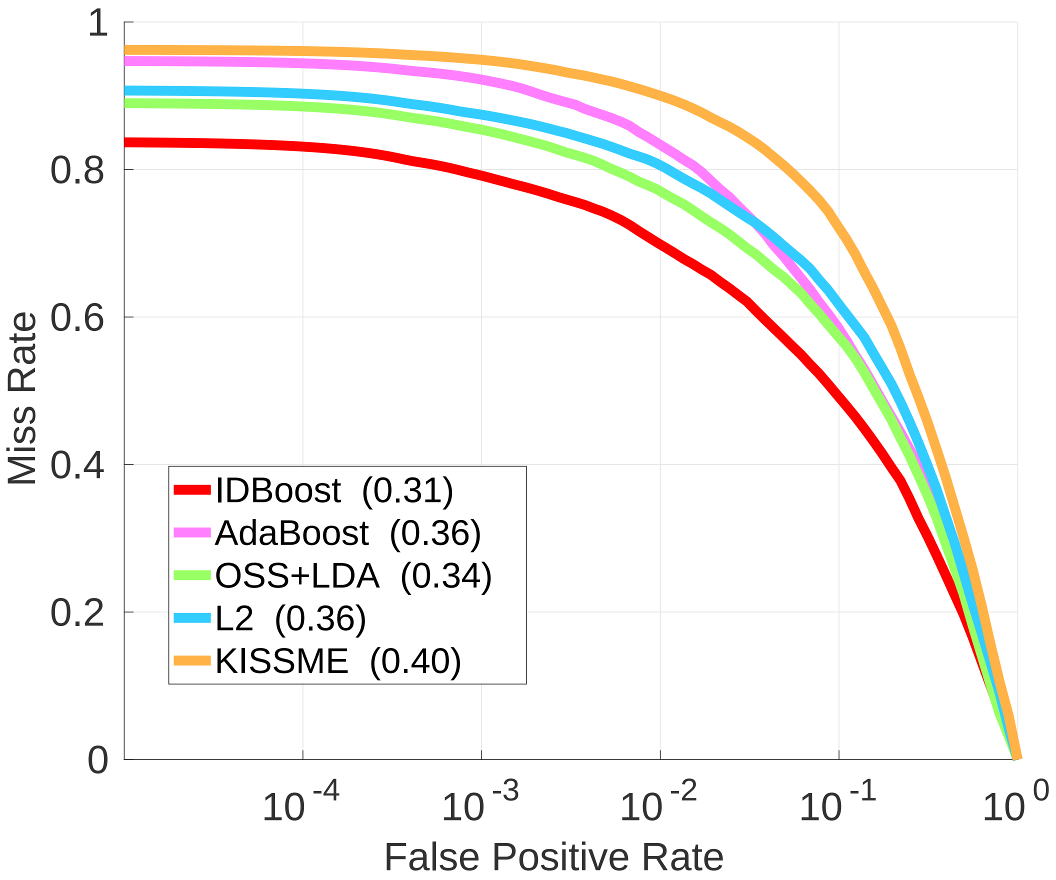


Figure 2.7: Reidentification performance. ROC curves for reidentification averaged over the 106 individuals in the RP dataset. The mean equal error rate is also reported. Again, IDBoost has the best mean equal error rate of 0.31. The closest competitor is again OSS+LDA. Performance on the RP dataset is 2.5 times worse than FPOQ indicating that the pedestrian dataset is far harder than the face dataset. This is not surprising since RP has a greater variation in pose and illumination than FPOQ. In the RP dataset there are ~2500 pedestrians detected per video.

subsets that IDBoost used to estimate σ. To then learn a KISSME individual detector at runtime, one of the inputs to the detector is fixed to be the feature vector of the fist instance of that individual.

The l^2 baseline detector operates in the same way as KISSME except that the metric is set to the identity.

Reidentification: Classification Performance

The results of the experiments are in Figures 2.6 and 2.7 with examples in Figure 2.5. They indicate that reidentification of faces is fairly easy due to the interview style of the videos with the pose, background and lighting of the face changing minimally so an individual looks the same from the first instance to its last. Reidentifying pedestrians is much more difficult due to the large changes in appearance that occur due to lighting and pose. From the ROC curves it is clear that our method (IDBoost) performs equally or better than any of the other methods for both datasets, despite the fact that its computational cost is a tiny fraction as shown in Figure 2.12 (top).

Unsurprisingly, the results also suggest that performance depends on the first instance of an individual used to train the detector. In Figure 2.5 we see that in the fifth row of faces the training example was selected during a fade in sequence of the video. As a result, the best scoring matches all have similar lighting rather than the same face.

Reidentification: Cross Dataset Perfomance

To evaluate the cross-dataset performance of IDBoost, experiments are also run on the VIPeR and ETHZ datasets (see Sec. 2.5 for details). Figures 2.8 and 2.9 show the results.

This experimental setup involves training IDBoost (the offline learning of σ (Eqn 2.4)) using the validation subset of RP and testing on the ETHZ and VIPeR datasets. For reference we provide the performance curves of a number of methods that were trained on those datasets[18, 41]. ID boost performs comparably to the other methods on the ETHZ dataset, whose statistics are close to RP's since individuals were sampled by a moving camera. On the VIPeR dataset, IDBoost performs less well than methods which were trained on VIPeR; this is probably because individuals were imaged by separate cameras with different lighting conditions.

For a fairer comparison, we also take the state-of-the-art method KISSME [32] and train it (estimate the covariance matrices offline (Sec. 2.6)) using the validation subset of RP. KISSME-RP performs poorly on both VIPeR and ETHZ. It over-fits the training data, so it is unable to generalise to new datasets with different statistics. This result suggests that IDBoost has a greater capacity to generalise across datasets.

Reidentification: Applications

Reidentification can be used in videos in a number of different ways. Here we demonstrate two possible applications where IDBoost is used as the underlying algorithm.

The first application is one where a user identifies an individual and they want to see all the frames of a video that contain that person. The identified user can be used

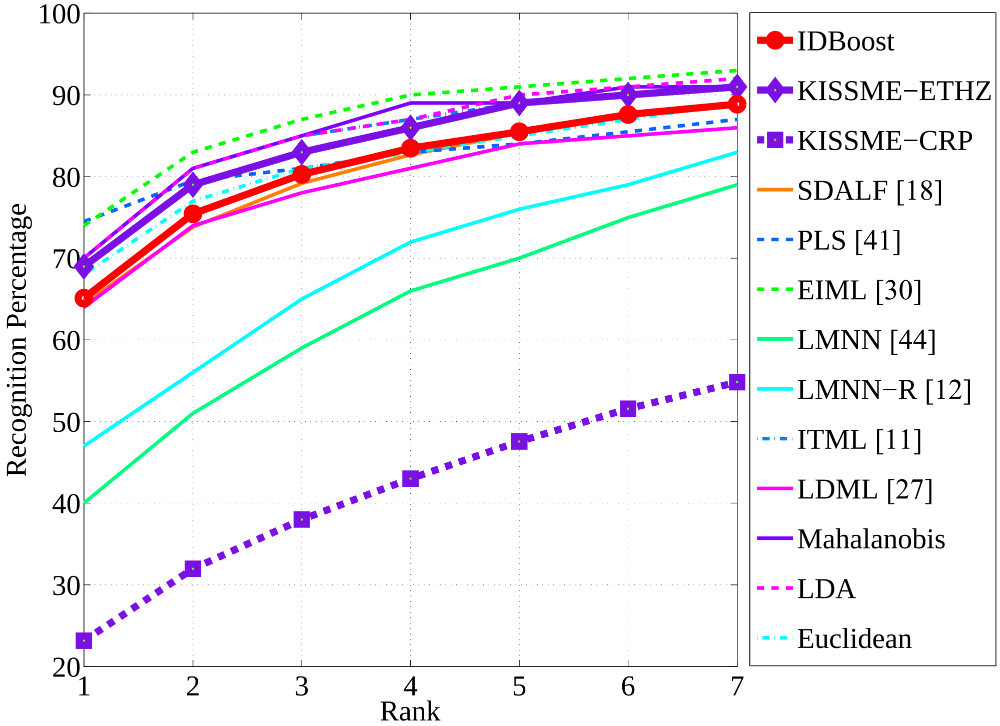


Figure 2.8: Cross-dataset performance of the proposed method (IDBoost) (red curves). The offline training of IDBoost and KISSME-RP was carried out using the validation subset of RP. The offline training of KISSME-ETHZ was carried out using the ETHZ SEQ2 training set (as were all the other methods shown whose curves have been provided as a reference [18, 41]). The ETHZ SEQ2 dataset was then used for testing. IDBoost performs significantly better than KISSME-RP, which suggests that IDBoost has a greater capacity to generalise across datasets. It also performs comparably to the other methods trained on the ETHZ dataset. This is because the statistics of ETHZ are simliar to CRP's since the dataset was collected by a moving camera.

to train an individual detector using IDBoost which can be applied to the remaining frames of the video. The system fast-forwards through the frames of the video that don't contain the person of interest. This can all be done in real time since the cost of training and operating the individual detector is low. In digital versions of this work, clicking on the left image in Figure 2.10 will redirect you to YouTube to view a demonstration of this application on a video from the FPOQ Dataset.

The second application is a reidentification system where a database of individuals is built as a video progresses. When a new individual is identified by the category detector an individual detector is learnt using that example. Each individual detector is then evaluated on the subsequent frames with the identity of the object being assigned to the best scoring detector (if the scores of all the individual detectors

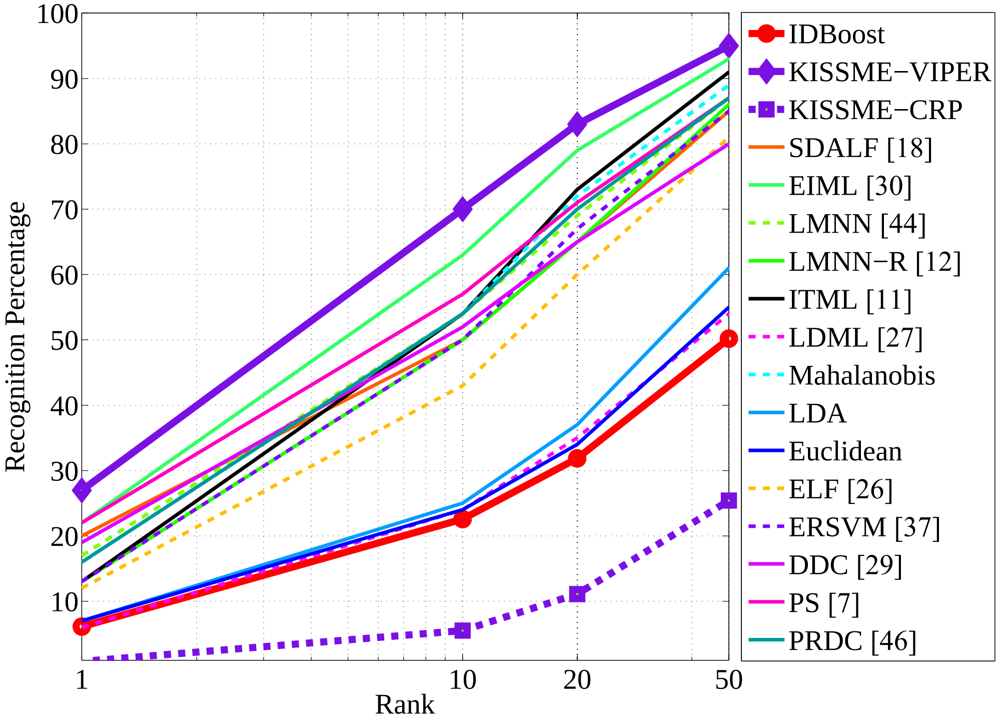


Figure 2.9: Cross-dataset performance of the proposed method (IDBoost) (red curves). The offline training of IDBoost and KISSME-RP was carried out using the validation subset of RP. The offline training of KISSME-VIPER was carried out using the VIPeR training set (as were all the other methods shown whose curves have been provided as a reference [18, 41]). The VIPeR dataset was then used for testing. IDBoost again performs better than KISSME-RP, however, when compared to other methods trained on the VIPeR dataset it does quite poorly. This is probably because the differences between RP and VIPeR are too great with VIPeR being collected from two disjoint cameras rather than from a moving vehicle.

are low, then the system would ask a human operator to verify if the example is a new individual). Even though the cost of running this system increases with the number of individuals, it is still very fast since each individual detector is only ever evaluated on windows in the image that the category detector has indicated contain an object of interest. In digital versions of this work, clicking on the right image in Figure 2.10 will redirect you to YouTube to view a demonstration of this application on a video from the FPOQ Dataset.

Tracking

The second mode of operation for an individual detector is as an actual sliding-window detector, where it is evaluated on every window in every frame of a video. The additional runtime cost in this mode of operation is higher than in the reidenti-

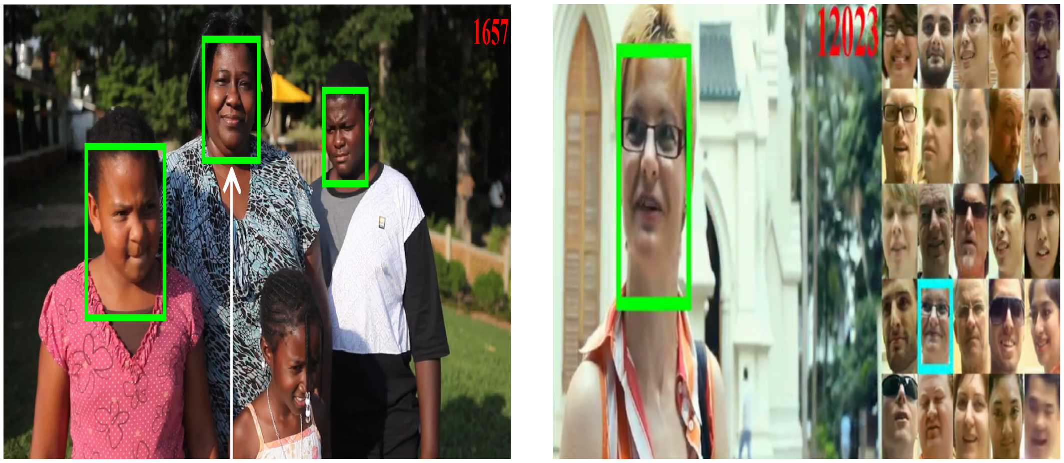


Figure 2.10: Demonstrations of reidentification applications using IDBoost. In digital versions of this work, clicking on the **(left)** image will redirect you to YouTube to view a demonstration of how frames containing certain individuals can be extracted from a video in the FPOQ Dataset. Clicking on the **(right)** image will take you to view a demonstration of how a database of individuals can be built as a video progresses.

fication scenario since individual detectors are now being applied to every window rather than just the windows that the category detector fires on. However, this extra cost is still very small since our method utilises the features that have already been computed by the category detector.

Experiments were carried out on both the FPOQ and the RP datasets to test this mode of operation. In this experiment sequences of individuals (a consecutive set of frames in which an individual occurs) were extracted and the category detector is evaluated on the first frame of the sequence. The output of the category detector is used to initialise an individual detector created using our method. The individual detector is then evaluated on the remaining frames in the sequence. This is a form of tracking by repeated detection using an appearance model. A motion model is not incorporated (it would be easy to implement this and would further reduce the additional runtime cost, but it would risk confusing the results of the experiments) and so the individual detector is evaluated on every window in a frame. Performance is measured by the number of times the tracker misses the individual it has been trained to track.

We compare the performance of our tracker to two other tracking methods: the Semi-Supervised Online Boosting Tracker [25] (OnlineBoost) (code obtained from authors website) and the Mean Shift or Kernel-based object tracker [9] (using the

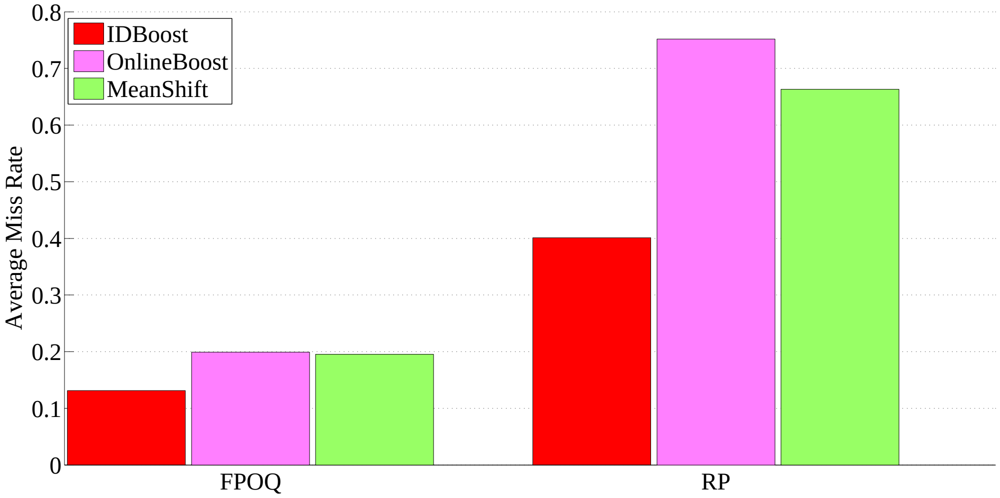


Figure 2.11: Tracking performance. The miss rate averaged over the (left) 680 sequences of individuals in the FPOQ dataset and over the (right) 199 sequences of individuals in the RP dataset. For each sequence, the output of the category detector evaluated on the first frame of the sequence is used to initialise an individual tracker (either an IDBoost, OnlineBoost or MeanShift tracker). The miss rate (the number of times the tracker misses the individual it was initialised to track) is then computed for each sequence and the average miss rate over all sequences is the computed. Our method has the best performance on the both datasets. In digital versions of this work, clicking on the plot will redirect you to YouTube to view a demonstration of three individuals being tracked from a video in the RP Dataset.

implementation in Dollar's toolbox). Both methods are initialised with the category detector output. Our method only uses the first instance of an individual from the first frame of a sequence whereas the other methods update the model of the individual over time.

The results in Figure 2.11 indicate that our method (IDBoost) has the best performance in terms of miss rate. Performance could be further improved by allowing the IDBoost tracker to update based on the appearance of the individual at the current frame rather than just using the appearance of the individual in the first frame of the sequence. Figure 2.12 (bottom) also shows that the additional computational cost of IDBoost is reasonable since real-time operation is still possible even without a motion model.

2.7 Discussion and Conclusions

We presented a method for training detectors of individual objects from a boosted category detector. Training happens in real-time using a single instance of an individual as a positive training example. The individual detectors make use of the

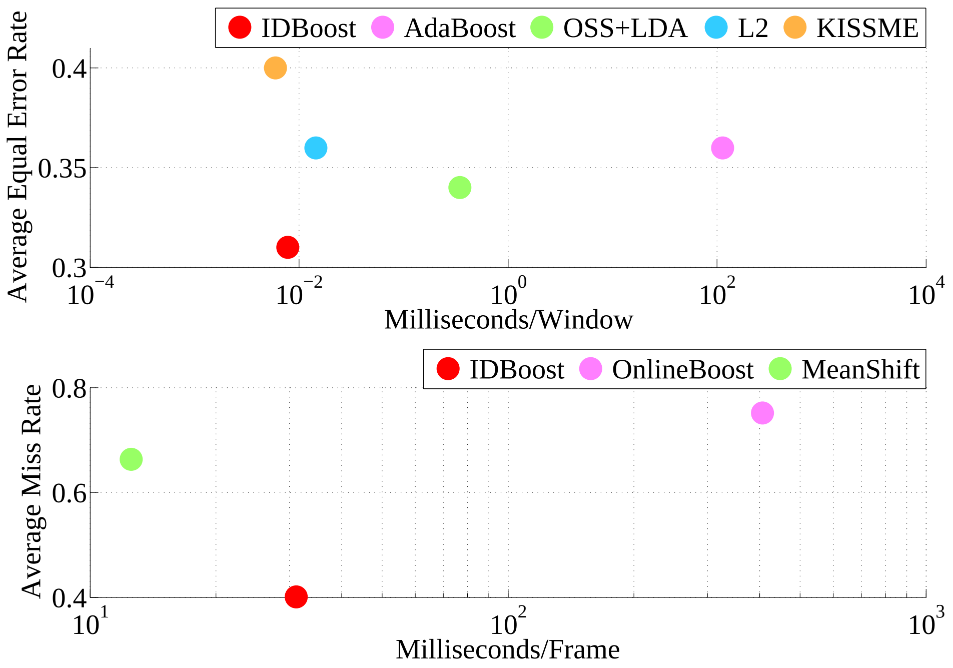


Figure 2.12: Computational performance. The average time it takes to train and evaluate an individual detector (either per window or per frame depending on the application) versus error rate for the (top) reidentification and (bottom) tracking scenarios using the RP dataset. Our method (IDBoost) operates just as fast as L2 and KISSME but has the best performance for the reidentification scenario. In the tracking scenario our method still achieves real-time operation with the best performance. This could be faster if a motion model was included. MeanShift is exceptionally fast for this reason but its performance is poor. All experiments were conducted on a single core of a 3.20 GHz processor. The ideal performance is in the bottom left corner of the plot.

category detector's feature computations; the thresholds for a single weak classifier are set using transfer learning. This ensures that the additional training and run-time costs for the individual detectors are minimal.

We carried out experiments on four datasets containing faces and pedestrians. The experiments were designed to test whether our simple and inexpensive strategy would work on real-world videos. We carried out two experiments. The first was designed to test reidentification, where the same individual is discovered across an entire video or image collection. The second was designed to test tracking, where an individual is tracked across consecutive video frames.

Our experiments suggest three conclusions: (a) both training and runtime computation of individual detectors is extremely inexpensive, (b) our method has both better tracking and reidentification performance than previous methods on the FPOQ and RP datasets, (c) the cross-dataset performance of our method is better than KISSME [32], a state-of-the-art, reidentification method.

Since our results show that individual object detectors can be trained quickly, it suggests that a tracking system robust to drift could be implemented. In this system, individual object detectors are used to track individuals and are updated using the appearance of the individual on a frame-by-frame basis rather than only using the first example of the individual, as is done in this work.

Acknowledgments

This work is funded by the Gordon and Betty Moore Foundation and the ONR MURI Grant N00014-10-1-0933.

References

[1] R. Appel et al. "Quickly Boosting Decision Trees-Pruning Underachieving Features Early". In: *ICML 2013*. 2013.

[2] R. Benenson et al. "Pedestrian Detection at 100 Frames per Second". In: *CVPR*. 2012. DOI: 10.1109/CVPR.2012.6248017.

[3] J. Berclaz et al. "Multiple Object Tracking using K-Shortest Paths Optimization". In: *PAMI* 33.9 (2011), pp. 1806–1819. DOI: 10.1109/TPAMI.2011.21.

[4] L. Bourdev and J. Brandt. "Robust Object Detection via Soft Cascade". In: *CVPR*. 2005. DOI: 10.1109/CVPR.2005.310.

[5] X. Burgos-Artizzu et al. "Merging Pose Estimates across Space and Time". In: *BMVC*. 2013. DOI: 10.5244/C.27.58.

[6] M. Burl, M. Weber, and P. Perona. "A Probabilistic Approach to Object Recognition Using Local Photometry and Global Geometry". In: *ECCV*. 1998. DOI: 10.1007/BFb0054769.

[7] D. S. Cheng et al. "Custom Pictorial Structures for Re-identification". In: *BMVC*. 2011. DOI: 10.5244/C.25.68.

[8] B. Coifman. "Vehicle Re-identification and Travel Time Measurement in Real-time on Freeways using Existing Loop Detector Infrastructure". In: *Transportation Research Record: Journal of the Transportation Research Board* 1643.1 (1998), pp. 181–191. DOI: 10.3141/1643-22.

[9] D. Comaniciu, V. Ramesh, and P. Meer. "Kernel-based Object Tracking". In: *PAMI* 25.5 (2003), pp. 564–577. DOI: 10.1109/TPAMI.2003.1195991.

[10] M. Cristani, S. Gong, and S. Yang, eds. *First International Workshop on Re-Identification*. Oct. 2012.

[11] J. Davis et al. “Information-Theoretic Metric Learning”. In: *ICML*. 2007. doi: 10.1145/1273496.1273523.

[12] M. Dikmen et al. “Pedestrian Recognition with a Learned Metric”. In: *ACCV*. 2010. doi: 10.1007/978-3-642-19282-1_40.

[13] P. Dollar, S. Belongie, and P. Perona. “The Fastest Pedestrian Detector in the West”. In: *BMVC*. 2010. doi: 10.5244/C.24.68.

[14] P. Dollár et al. “Integral Channel Features”. In: *BMVC*. 2009. doi: 10.5244/C.23.91.

[15] A. Ess, B. Leibe, and L. Van Gool. “Depth and Appearance for Mobile Scene Analysis”. In: *ICCV*. 2007. doi: 10.1109/ICCV.2007.4409092.

[16] M. Everingham and et al. “The 2005 PASCAL Visual Object Classes Challenge”. In: *First PASCAL Machine Learning Challenges Workshop, MLCW*. 2005, pp. 117–176. doi: 10.1007/11736790_8.

[17] M. Everingham, J. Sivic, and A. Zisserman. “Hello! My name is... Buffy – Automatic Naming of Characters in TV Video”. In: *BMVC*. 2006. doi: 10.5244/C.20.92.

[18] M. Farenzena et al. “Person Re-Identification by Symmetry-Driven Accumulation of Local Features”. In: *CVPR*. 2010. doi: 10.1109/CVPR.2010.5539926.

[19] P. Felzenszwalb et al. “Object Detection with Discriminatively Trained Part-Based Models”. In: *PAMI* 32.9 (2010), pp. 1627–1645. doi: 10.1109/TPAMI.2009.167.

[20] R. Fergus, P. Perona, and A. Zisserman. “Object Class Recognition by Unsupervised Scale-invariant Learning”. In: *CVPR*. 2003. doi: 10.1109/CVPR.2003.1211479.

[21] M. Fischler and R. Elschlager. “The Representation and Matching of Pictorial Structures”. In: *IEEE Transactions on Computers* 22 (1973), pp. 67–92. doi: 10.1109/T-C.1973.223602.

[22] Y. Freund and R. E. Schapire. “A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting”. In: *Journal of Computer and System Sciences* 55.1 (1997), pp. 119–139. doi: 10.1006/jcss.1997.1504.

[23] H. Grabner, M. Grabner, and H. Bischof. “Real-time Tracking via On-line Boosting”. In: *BMVC* (2006). doi: 10.5244/C.20.6.

[24] H. Grabner and H. Bischof. “On-line Boosting and Vision”. In: *CVPR*. 2006. doi: 10.1109/CVPR.2006.215.

[25] H. Grabner, C. Leistner, and H. Bischof. “Semi-supervised On-line Boosting for Robust Tracking”. In: *ECCV*. 2008. doi: 10.1007/978-3-540-88682-2_19.

[26] D. Gray and H. Tao. "Viewpoint Invariant Pedestrian Recognition with an Ensemble of Localized Features". In: *ECCV*. 2008. DOI: 10.1007/978-3-540-88682-2_21.

[27] M. Guillaumin, J. Verbeek, and C. Schmid. "Is that you? Metric Learning Approaches for Face Identification". In: *ICCV*. 2009. DOI: 10.1109/ICCV.2009.5459197.

[28] O. Hamdoun et al. "Person Re-identification in Multi-camera System by Signature based on Interest Point Descriptors Collected on Short Video Sequences". In: *ICDSC*. 2008. DOI: 10.1109/ICDSC.2008.4635689.

[29] M. Hirzer, P. Roth, and H. Bischof. "Person Re-identification by Efficient Impostor-Based Metric Learning". In: *AVSS*. 2012. DOI: 10.1109/AVSS.2012.55.

[30] M. Hirzer et al. "Person Re-identification by Descriptive and Discriminative Classification". In: *Scandinavian Conference on Image Analysis*. 2011. DOI: 10.1007/978-3-642-21227-7_9.

[31] Z. Kalal, K. Mikolajczyk, and J. Matas. "Tracking-Learning-Detection". In: *PAMI* 34.7 (2012), pp. 1409–1422. DOI: 10.1109/TPAMI.2011.239.

[32] M. Kostinger et al. "Large Scale Metric Learning from Equivalence Constraints". In: *CVPR*. 2012. DOI: 10.1109/CVPR.2012.6247939.

[33] L. Wolf, T. Hassner, and Y. Taigman. "The One-Shot Similarity Kernel". In: *ICCV*. 2009. DOI: 10.1109/ICCV.2009.5459323.

[34] M. Lades et al. "Distortion Invariant Object Recognition in the Dynamic Link architecture". In: *IEEE Transactions on Computers* 42.3 (1993), pp. 300–311. DOI: 10.1109/12.210173.

[35] Y. Lecun et al. "Gradient-Based Learning Applied to Document Recognition". In: *Proceedings of the IEEE* 86.11 (Nov. 1998), pp. 2278–2324. DOI: 10.1109/5.726791.

[36] D. G. Lowe. "Distinctive Image Features from Scale-Invariant Keypoints". In: *Int. J. Comput. Vision* 60.2 (2004), pp. 91–110. DOI: 10.1023/B:VISI.0000029664.99615.94.

[37] A. Mignon and F. Jurie. "PCCA: A New Approach for Distance Learning from Sparse Pairwise Constraints". In: *CVPR* (2012). DOI: 10.1109/CVPR.2012.6247987.

[38] B. Prosser et al. "Person Re-Identification by Support Vector Ranking". In: *BMVC*. 2010. DOI: 10.5244/C.24.21.

[39] D. Ramanan et al. "Tracking People by Learning Their Appearance". In: *PAMI* 29.1 (2007), pp. 65–81. DOI: 10.1109/TPAMI.2007.250600.

[40] M. Riesenhuber, T. Poggio, et al. "Hierarchical Models of Object Recognition in Cortex". In: *Nature Neuroscience* 2 (1999), pp. 1019–1025.

[41] P. M. Roth et al. “Mahalanobis Distance Learning for Person Re-identification”. In: *Person Re-Identification*. 2014. Chap. 12, pp. 247–267. doi: 10.1007/978-1-4471-6296-4_12.

[42] W. Schwartz and L. Davis. “Learning Discriminative Appearance-Based Models Using Partial Least Squares”. In: *CGIP*. 2009. doi: 10.1109/SIBGRAPI.2009.42.

[43] P. Viola and M. Jones. “Rapid Object Detection using a Boosted Cascade of Simple Features”. In: *CVPR*. 2001. doi: 10.1109/CVPR.2001.990517.

[44] P. Viola and M. J. Jones. “Robust Real-Time Face Detection”. In: *Int. J. Comput. Vision* 57.2 (2004), pp. 137–154. doi: 10.1023/B:VISI.0000013087.49260.fb.

[45] K. Q. Weinberger and L. K. Saul. “Fast Solvers and Efficient Implementations for Distance Metric Learning”. In: *ICML*. 2008. doi: 10.1145/1390156.1390302.

[46] W.-S. Zheng, S. Gong, and T. Xiang. “Person Re-Identification by Probabilistic Relative Distance Comparison”. In: *CVPR*. 2011. doi: 10.1109/CVPR.2011.5995598.

Chapter 3

TRACKING

The contents of this chapter are from the peer-reviewed publication “Online, Real-Time Tracking using a Category-to-Individual Detector” by D. Hall and P. Perona, appearing at ECCV 2014[1].

3.1 Abstract

A method for online, real-time tracking of objects is presented. Tracking is treated as a repeated detection problem where potential target objects are identified with a pre-trained category detector and object identity across frames is established by individual-specific detectors. The individual detectors are (re-)trained online from a single positive example whenever there is a coincident category detection. This ensures that the tracker is robust to drift. Real-time operation is possible since an individual-object detector is obtained through elementary manipulations of the thresholds of the category detector and therefore only minimal additional computations are required. Our tracking algorithm is benchmarked against nine state-of-the-art trackers on two large, publicly available and challenging video datasets. We find that our algorithm is 10% more accurate and nearly as fast as the fastest of the competing algorithms, and it is as accurate but 20 times faster than the most accurate of the competing algorithms.

3.2 Introduction

The objective of tracking is to determine the size and location of a target object in a sequence of video frames, given the initial state of the target. It is important for a variety of applications, including the tracking of pedestrians in railway stations and airports, vehicles on the road for traffic monitoring, and faces for interfacing people with computers. It is a well studied problem and although many offline tracking methods exist [3, 9, 10, 30, 33, 34, 37, 38], the focus of this work will be on online trackers.

We present a novel method for real-time, online, appearance-based tracking. Tracking is treated as a detection problem where an individual-object detector is trained online to distinguish the target-to-be-tracked from background. Objects to track are

[1] Project Website: http://vision.caltech.edu/~dhall/projects/CIT/

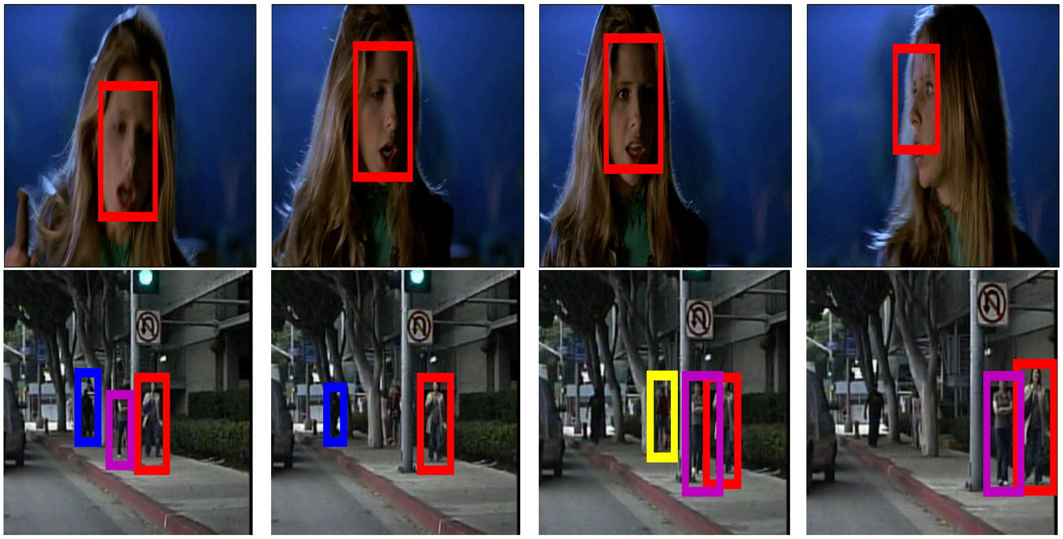


Figure 3.1: Tracking individuals across a video sequence using our proposed tracking algorithm. **(Top)** A face from the Buffy dataset [41] and **(bottom)** pedestrians from Caltech Pedestrians [17, 18]. The appearance of the face changes over time with a full frontal face example at initialisation (far left) to a quasi-frontal face in the final frame (far right). Despite the change in appearance, our tracker is able to track the target since the model for the target is updated over time (see Sec. 3.4). Pedestrians are also tracked successfully even when they are subject to occlusion. The magenta target is initialised in the first frame, occluded by the light pole in the second and successfully reacquired in the third. Individual tracker outputs are colour-coded. Each image is 30 frames apart (model update is still occurring every frame).

first identified with a pre-trained category detector (a face, pedestrian or vehicle detector for example); each of the detections made by the category detector are now identified as individual targets to be tracked; for each of these targets an individual detector is learnt on-the-fly using IDBoost, a category-to-individual learning algorithm [2]. During tracking, the individual detector is updated using the currently detected sample of the target but only if there is a coincident category detection. This ensures that the tracker is robust to drift. Crucially, the algorithm runs in real-time since the individual-object detector is obtained through elementary manipulations of the thresholds of the category detector. We then benchmark our tracking algorithm against nine other publicly available, state-of-the-art trackers on two challenging video datasets. We make two main contributions:

1. A fast, accurate, tracking algorithm that is robust to drift.

2. A careful and reliable benchmark of state-of-the-art trackers.

3.3 Related work

Appearance-based tracking algorithms typically contain the following elements: 1) an appearance model for the target object to be tracked, 2) a search strategy to find potential candidates in subsequent frames that match the target and 3) a mechanism to dynamically update the appearance model of the target so that changes in pose and illumination over time can be modelled.

A pitfall of dynamically updating the appearance model is that when trackers make mistakes this incorrect information is then incorporated into the model. The result is that the tracker no longer tracks the original target. This is known as drift. Designing algorithms that are robust to noisy updates is thus essential for good tracking performance.

Many trackers use generative appearance models. One of the representations used for target objects within this class of trackers are subspaces. Black and Jepson [12] learn offline an eigenbasis to model the target object along with particle filtering as a search mechanism. The IVT method of Ross *et al*. [39] proposes an online update of the target subspace over time using incremental PCA. Wu's ORIA [35] tracker also includes online updates but updates only occur if the new target is significantly different from the existing basis set.

Kernel based methods are also used to represent target objects. The influential work of Comaniciu *et al*. with their Kernel-Based Object Tracker (KMS) [14] represent the target object with a histogram in some feature space; a metric based on the Bhattacharyya coefficient is used to match the target to potential candidates, while the mean-shift algorithm [24] is used to efficiently search for these candidates. In traditional histogram-based algorithms information about the spatial distribution of features is lost; the FRAG [1] tracker of Adam *et al*. represents the target using multiple histograms obtained from many different patches in the target, thus preserving some of the spatial information. Neither of these methods update the model online.

Alternative representations include probability distribution fields (DFT) [40], sparse linear combinations of target and trivial templates (L1AP) [6] and the superpixels of (LOT) [36].

Discriminative appearance models are also widely used. Avidan's ensemble tracker [4] constructs a feature vector for every pixel. A classifier to separate pixels belonging to the target from those in the background is then trained by using an adaptive ensemble of weak classifiers. The compressive tracking (CT) algorithm of Zhang *et al*. [46]

generates multi-scale features for the positive and negative samples and applies a sparse sampling matrix to reduce dimensionality. A naive Bayes classifier with online update is then used to classify windows as target or background. Grabner *et al*.'s online boosting method (OAB) [25] adaptively selects features to discriminate the object from the background.

To avoid drift, Grabner *et al*. [26] propose a semi-supervised, online boosting algorithm (SBT) where only the initial samples of the target are labelled while all of the self-learnt samples are unlabelled. The MIL tracker of Babenko *et al*. [5] uses multiple instance learning where a bag of positive samples are used to update the model. Kalal's [31] TLD tracker also approaches tracking as a semi-supervised learning problem with positive and negative examples being selected by an online classifier that has structural constraints. The BSBT tracker of Stalder *et al*. [42] combines the supervised and semi-supervised approaches into a single implementation.

For all of the approaches mentioned so far a sparse sampling strategy is used. This means that in each frame, positive samples are collected close to the predicted target while the negative samples are further from the target's centre. The CSK algorithm of Henriques *et al*. [28] proposes a different approach where a classifier is trained using all possible samples in a dense sampling strategy. The circulant structure of the problem is exploited allowing for not only efficient training but fast detection since all responses can be computed simultaneously rather than using a sliding-window scheme.

There are many benchmark datasets available for a number of vision problems. For pedestrians there is INRIA [15] and Caltech Pedestrians [17], for unconstrained face recognition there is LFW [29], and for person reidentification there is VIPeR [27]. Tracking, however, still lacks a decent benchmarking dataset although progress has been made recently to fill this gap. Wu *et al*. [45] have collected a benchmark that contains 50 sequences commonly used in the literature to evaluate tracking algorithms. While most algorithms have been evaluated on a subset of these sequences by their original authors, Wu *et al*. provide a far more comprehensive analysis by evaluating 29 tracking algorithms on all 50 sequences given the initial bounding box of the target object.

While the progress that has been made by Wu *et al*. is appreciable, we feel that the dataset is lacking for the following reasons: 1) the size of the dataset is too small; 2) the difficulty of most sequences is low with a focus on tracking only single objects; 3) about half of the sequences are unrealistic as they are in controlled environments;

and 4) trackers are perfectly initialised by a ground truth bounding box. This gives little insight into how robust trackers are to poor initialisation. Wu *et al.* address this by jittering the ground truth bounding box.

3.4 Approach

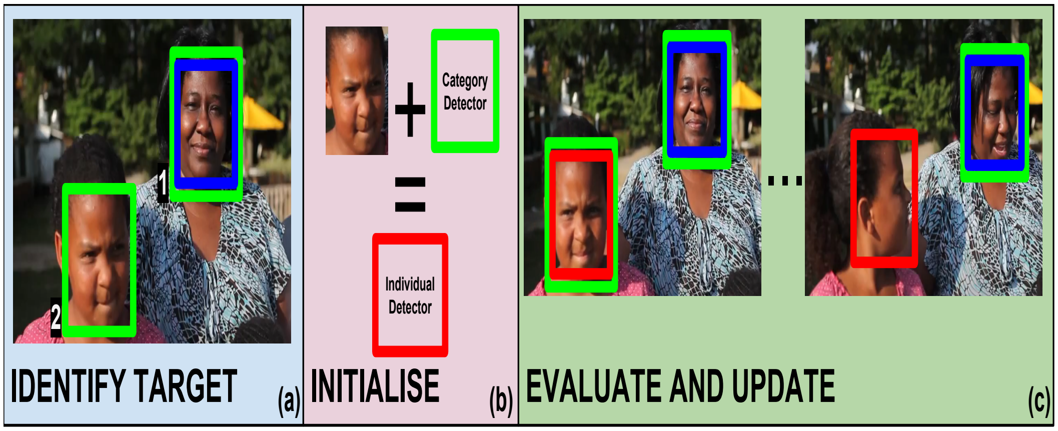


Figure 3.2: Tracking Algorithm Outline. (a) A new target (2) is identified by a non-coincident category detection (green). There is also a category detection that coincides with an individual detection (blue) which indicates that this target (1) is already being tracked. (b) An individual detector (red) for target 2 is initialised using the category-to-individual learning algorithm in [2]. The only two pieces of information required to initialise the individual detector is a single positive example of the target and the category detector itself. (c) The individual detector is then evaluated on subsequent frames using a sliding window scheme. The location with the maximal score is identified as the new location of the target object. If a detection made by the individual detector is coincident with a detection made by the category detector then the individual detector is re-initialised with the current example of the target. If there is no coincident category detection the individual detector is not updated. If this occurs for more than a fixed number of frames tracking of that target stops.

In this section we present the details of our tracking algorithm. We treat the tracking problem as a detection task and train an individual-object detector, online, to distinguish the target from background. We break down the algorithm into 5 major components: 1) identify target objects to track; 2) initialise the tracker; 3) evaluate the tracker on a new frame; 4) update the tracker; 5) stop the tracker. There is no assumption made about the number of objects being tracked and our algorithm is able to handle tracking multiple objects that enter and exit a scene. An outline of our approach is depicted in Figure 3.2.

Identify Target Objects to Track

Identifying new targets to track is a problem that most of the tracking literature avoids. It is usually assumed that an initial tracking window has already been provided either manually or by a 'perfect' category detector [39]. In this work, instead of providing initial locations by hand, a category detector is used. This is a more realistic setting under which trackers would operate since for most online applications, having a human operator identify potential targets would not be feasible, particularly if there are many targets to identify. This setting also allows us to evaluate how robust tracking algorithms are to poor initialisation.

A new target is identified if the category detector makes a detection and there is no coincident individual detection as shown in Figure 3.2.

Category detectors are trained offline using AdaBoost [23]. A boosted classifier takes feature vector $\mathbf{x} \in \mathbb{R}^D$ as input and outputs a binary decision:

$$H(\mathbf{x}) = \text{sign}\left(\sum_{m=1}^{M} \alpha_m h_m(\mathbf{x}) - \tau\right) \tag{3.1}$$

where $h_m(\mathbf{x})$ is a weak classifier; α_m its weight and the threshold τ is chosen to produce the desired trade-off between false reject rate and false alarm rate.

The family of weak classifiers used are stumps. This means that given an input $\mathbf{x} \in \mathbb{R}^D$, the decision only depends on the j-th dimension of $\mathbf{x}$, a threshold $\theta \in \mathbb{R}$ and a polarity $p \in \{\pm 1\}$

$$h_m(\mathbf{x}) = \begin{cases} 1, & p_m x_{j_m} > p_m \theta_m \\ -1, & \text{otherwise} \end{cases} \tag{3.2}$$

Note that any boosting method and decision trees of any depth could be used to train the category detector; our proposed method is agnostic to these choices.

Initialise Tracker

Once a new target has been identified; a tracker can now be initialised to track the object. In this section we briefly outline IDBoost, the category-to-individual learning algorithm (see [2] for details), which allows us to efficiently train the individual detector that only detects the target object.

The approach for learning an individual detector from a category detector has four elements:

The individual detector is a boosted cascade of classifiers. Cascades are fast and their performance is state-of-the-art [8, 20, 16, 43, 44], making the individual detector suitable for real-time operation.

The individual detector is learnt from a single sample. The object identified by the category detector, which we will denote by $\mathbf{u}^0 \in \mathbb{R}^D$, is used as the single positive training example to train the individual detector. Costly computations are avoided by using a single positive sample and by not mining negative samples.

The individual detector uses the same M features that were selected by AdaBoost for the category detector. We will denote this set of features by $\mathbf{J} = (j_1, \ldots, j_M)$ where $j_m \in \{1, \ldots, D\}$ and the importance of each feature through the weights $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_M)$. Computational cost is reduced since there is no need to compute extra features (this has already been done for the category detector). There is also no cost for selecting which features to use since they are fixed.

Only the thresholds for a single weak classifier in the boosted cascade of the individual detector are modified. Selecting the thresholds for a single weak classifier $h'(x')$, which depends on a single feature $x' \in \mathbb{R}$, can be achieved at almost zero computational cost.

Consider the target that has been detected by the category detector and call u' the value of feature x'. An interval, centred at u', can now be defined. The width of this interval is determined offline, from a small validation set that contains tracks of a few individuals. The standard deviation of feature x' for a single individual across its track is calculated; the median standard deviation or spread σ' is then computed across the set of individuals. The spread σ' gives an estimate of the width of the interval.

Formally, the interval is $(u' - \beta\sigma', u' + \beta\sigma')$ where β is a free parameter that can be tuned experimentally. If the appearance of the individual is changing slowly over time, which is a reasonable assumption to make for video (since the difference from frame-to-frame is small), this interval represents the most likely values that the feature x' will take for that individual.

The weak classifier $h'(x')$ can thus be obtained from one training example and by only setting two thresholds (making the computational cost near zero):

$$h'(x'; u', \sigma') = \begin{cases} 1 & u' - \beta\sigma' < x' < u' + \beta\sigma' \\ -1 & \text{otherwise.} \end{cases} \tag{3.3}$$

This weak classifier provides evidence for an individual being present (absent) if the feature x' lies inside (outside) the interval $(u' - \beta\sigma', u' + \beta\sigma')$.

An individual detector in the form of a cascaded boosted classifier can now be constructed. Given the set of features $\mathbf{J}$ and weights $\boldsymbol{\alpha}$ that were selected for the category detector, the single positive example $\mathbf{u}^0$ of the target, and an estimate of the spread $\boldsymbol{\sigma} = (\sigma_1, \ldots, \sigma_M)$ of the features $\mathbf{J}$, a classifier $F(\mathbf{x})$ for the target can be defined by:

$$F(\mathbf{x}; \mathbf{u}^0, \boldsymbol{\sigma}) = \sum_{m=1}^{M} \alpha_m h'(x_{j_m}; u_{j_m}, \sigma_m) \tag{3.4}$$

Evaluate Tracker

Given the next frame in the video the individual detector $F(\mathbf{x}; \mathbf{u}^0, \boldsymbol{\sigma})$ is evaluated using a sliding-window scheme. The location with the maximal classification score is identified as the new location of the target object. Since the individual detector is a cascade of boosted classifiers, sliding window detection is very efficient. It would also be possible to use a motion model to reduce the number of sub-windows evaluated (we have not done this).

Update Tracker

If at the new location of the target object (the one identified by the individual detector) there is also a coincident detection made by the category detector, then the individual detector is updated with the new sample, $\mathbf{u}^1$, of the target. The update procedure is then as simple as reinitialising the individual detector with the new sample which results in $F(\mathbf{x}; \mathbf{u}^1, \boldsymbol{\sigma})$. The updated individual detector is then applied to the next frame and so on. If there fails to be a coincident category detection then the individual detector is not updated at all and is simply applied to the next frame. Figure 3.2 outlines this update procedure. Drift is avoided by only updating the model when the individual detector and category detector are coincident. This strategy ensures that only “good” positive samples are used to update the model. A category detection and individual detection are coincident if their overlap is greater than 50%.

Stop Tracker

There are two conditions, either of which can be met, for tracking of the target object to cease. The first is that there are no detections made by the individual detector for T_1 consecutive frames. This condition is usually met when the target object leaves the frame. The second is that the detections made by the individual detector fail to coincide with those made by the category detector for T_2 consecutive frames. This condition is usually met when a tracker is initialised by a false detection made by the category detector. In this work we set both of these quantities to five frames.

3.5 Datasets

To benchmark the performance of our tracking algorithm we use two publicly available video datasets.

The first dataset is the Caltech Pedestrians [17, 18] dataset. It contains 250,000 frames of video, at a resolution of 640x480, taken from a vehicle driving through regular traffic in an urban environment. The dataset is labelled with a total of 350,000 bounding boxes and around 1900 unique individual pedestrians. Around 30% of the frames have two or more pedestrians. Pedestrians are visible for 150 frames on average. The dataset is divided into 11 sets; 6 are used for training (S0-S5) and 5 are used for testing (S6-S10). This division is provided by the authors. The training set contains 192,000 bounding boxes and the test set contains 155,000. The authors also refer to experiments conducted on pedestrians over 50 pixels tall, with no or partial occlusion as the *reasonable evaluation setting*. There are 73,256 labelled bounding boxes and 769 unique individuals that meet these requirements in the test set. All of our experiments are conducted using the reasonable evaluation setting.

The second dataset used is the Buffy dataset [41]. It has been regularly used for the automatic labelling of faces of TV characters using subtitle and script text [41, 21]. In this work we use the publicly available, ground truth labels of [7]. The dataset contains episodes 1–6 from season 5 of Buffy the Vampire Slayer with around 64,000 frames per episode. The faces are labelled using an automatic algorithm rather than by a human operator. This means that the ground truth labels are noisy. In total there are 317,831 labelled bounding boxes and 5513 unique face tracks across the six episodes. There is also a wide variety in the appearance of individuals in this dataset with many shots set outdoors and at night time; there are also a number of close-up shots.

Examples of the different individuals and how their appearance changes over time for both datasets are displayed in Figure 3.3.

The reason we benchmark tracking performance on these datasets is 1) because of their size and 2) because of their complexity. The benchmarking procedure of

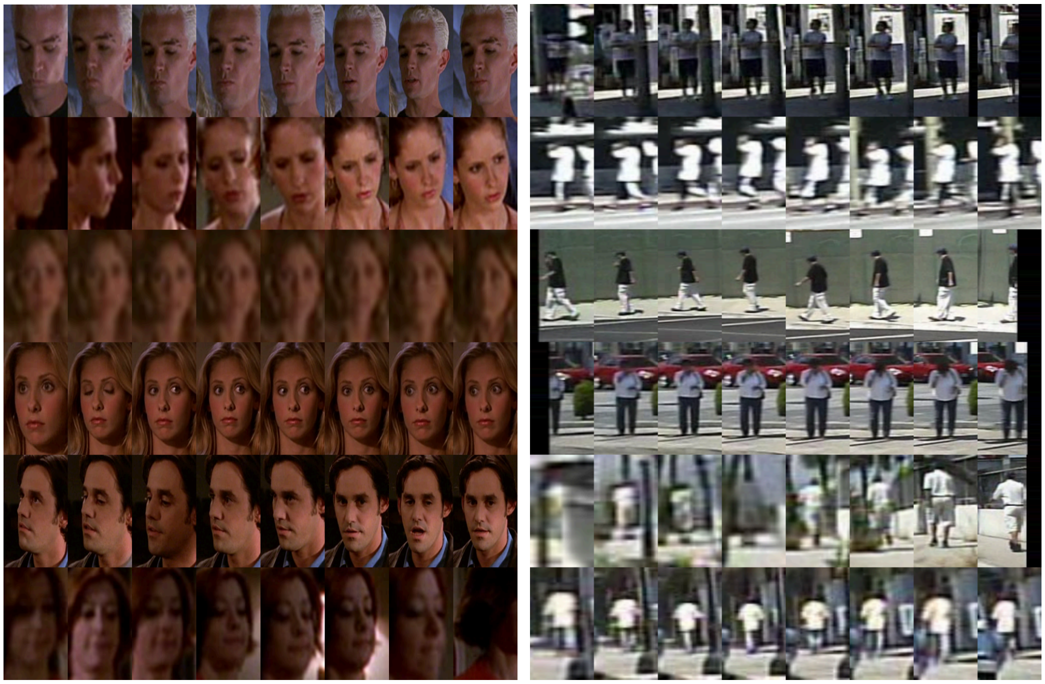


Figure 3.3: Dataset Examples. **(Left)** face tracks from the Buffy dataset and **(right)** pedestrian tracks from Caltech Pedestrians. The ground-truth trajectories were sampled randomly from a video in each of the datasets. There are frontal and profile faces; the lighting can change considerably across a face track; facial expressions are varied. The pedestrians are low resolution; they change scale as the car mounted camera approaches and there is occlusion. The datasets are large with a combined total of 7500 individual tracks and 800,000 bounding boxes. They are also difficult due to the variety in pose, lighting, background and scale across a track as well as having multiple objects present at any one time. To the best of our knowledge this is one of the largest datasets that appearance-based tracking algorithms have been evaluated on. In digital versions of this work, clicking on the either of the images will redirect you to YouTube to view one of the videos in the corresponding dataset.

Wu *et al*. [45] only evaluates tracking performance on 50 different individuals. In this work we make this evaluation on around 6000 individuals, a 120-fold increase. The datasets here are also more complex. Having to track multiple objects of a similar appearance is a far more difficult task than the tracking of a single object, particularly when the target objects interact with each other, which may lead to trackers swapping identities. These datasets are also less biased since they were collected independently of the authors of any the trackers mentioned in section 3.2. They are also more realistic in the sense that the sequences haven't been generated in a lab as around half of the sequences in [45] are.

3.6 Performance Metrics

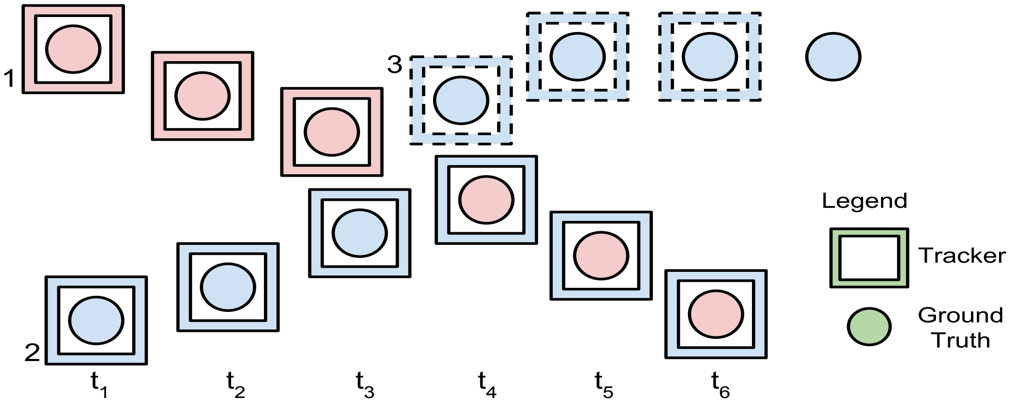


Figure 3.4: A graphical example of the performance measures. The performance measures used are precision, recall and continuity/fragmentation of a trajectory. Tracker 1 (pink squares) is initialised to track the ground-truth target *A* (pink circles). Tracker 1 has 3 matches since it is only ever matched with target *A* and has a track length of 3. Tracker 2 (solid blue squares) only has 3 matches with a length of 6. This is because it is initialised to track the ground-truth target *B* (blue circles), however, it's last three detections coincide with ground-truth target *A* which is incorrect. Tracker 3 (dotted blue squares) is also initialised to track ground-truth target *B* and has 3 matches with a length of 3. The precision is then 9/12. Ground-truth target *A* has 3 matches with a length of 6. The last three ground-truth targets are unmatched due to tracker 2 being initialised to track target B. Ground-truth target *B* has 6 matches and a length of 7 since all of the corresponding trackers were initialised to track target *B*. The recall is 9/13. Target *A* is only covered by tracker 1 (tracker 2 is not included since it has not been initialised to track *A*) while target *B* is covered by trackers 2 and 3. The fragmentation is thus 3/2 and so the continuity is 2/3.

There are a variety of methods [45, 32, 11] to measure the performance of a tracking algorithm. In this work we use three simple and intuitive measures to capture tracking performance (refer to Figure 3.4).

In a multiple-object tracking scenario, given a single frame, there are a number of options for determining whether a tracked target and ground-truth location match. The centre location error which is the Euclidean distance between the centre's of the two locations is one option [5, 45, 11]. This measure is not very robust to labelling error and relies heavily on the fact that the ground-truth is perfect. A more robust method to use is the bounding box overlap. Using the PASCAL criteria [22] a tracked object matches a ground-truth object if the area of overlap between their respective bounding boxes exceeds 50%.

In addition to bounding box overlap we also include another criterion for matching to occur. A tracked object matches a ground-truth object if the tracker was initialised by a target that has the same identity as the ground-truth object. This is a reasonable condition to enforce since a tracker initialised by target A should not be tracking target B.

Now that single frame matching has been defined, performance across trajectories can be measured. If there are a total of N tracking trajectories and M ground truth trajectories then three quantities can be defined: precision, recall and continuity.

If $l^n_{tracker}$ is the length of tracking trajectory n and $d^n_{tracker}$ is the number of matches in trajectory n, then the precision P across all tracking trajectories is defined as:

$$P = \frac{\sum_{n=1}^{N} d^n_{tracker}}{\sum_{n=1}^{N} l^n_{tracker}} \tag{3.5}$$

This measure is similar to the MOTP metric proposed by Bernardin and Stiefelhagen [11]. Precision gives a measure of how well a tracker initialised on target A tracks target A. If the tracker drifts or there is an identity swap, precision will be low.

If l^m_{gt} is the length of ground truth trajectory m and d^m_{gt} is the number of matches in ground truth trajectory m then the recall R across all ground-truth trajectories is defined as:

$$R = \frac{\sum_{m=1}^{M} d^m_{gt}}{\sum_{m=1}^{M} l^m_{gt}} \tag{3.6}$$

Let f^m be the number of trackers that are required to cover ground truth trajectory m. The continuity C is then defined as:

$$C = \frac{\sum_{m=1}^{M} \mathbb{1}(f^m > 0)}{\sum_{m=1}^{M} f^m} \tag{3.7}$$

where $\mathbb{1}$ is the indicator function. Continuity is the inverse of the mean number of trackers needed to cover a ground truth trajectory. This is described as fragmentation in [32]. Trajectories that have no trackers covering them are not included in this calculation. The reason continuity is used instead of fragmentation is that continuity is easier to compare to precision and recall since a value of 1 for all three of these measures indicates perfect performance.

Recall and continuity give a measure of how well targets are tracked. If the tracker does not adapt to the appearance of the target, continuity will be low.

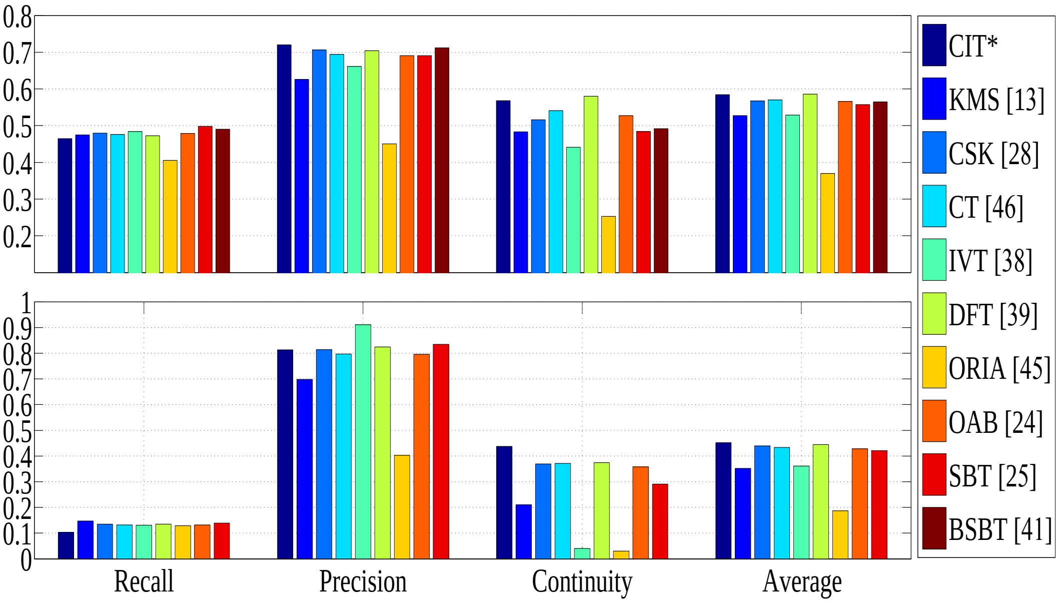


Figure 3.5: Tracking Performance on Buffy **(top)** and Caltech Pedestrians **(bottom)** for a category detector operating at a recall of 0.5. Refer to Section 3.6 for definitions of recall, precision and continuity. The average performance is the mean of the precision, recall and continuity. The precision calculation ignores trajectories that have been initialised by false category detections. A tracker with perfect performance would have a value of 1 for each of the measurements. The results indicate that our method CIT is one of the best performers; however, most of the methods have similar performance results. The relative performance of the trackers is roughly equivalent across the two very different, very distinct datasets. The BSBT algorithm ran too slowly and so its performance on Caltech Pedestrians is omitted

The overall performance of a tracking algorithm is given as the mean of the precision, recall and continuity. We feel that this is a reasonable metric to compare algorithms on since all three quantities are equally as important.

3.7 Experiments

For the remainder of this paper we will refer to our proposed tracking algorithm as the Category-to-Individual Tracker (CIT). To assess the performance of CIT we conduct experiments using the Buffy and Caltech Pedestrian datasets (refer to Sec 3.5). To the best of our knowledge, this is one of the largest performance evaluations for appearance-based tracking. The most important findings are reported here in this manuscript; the remainder are included as supplementary material in Sec. 3.9.

The results of CIT are compared to 9 other, publicly available, appearance-based tracking algorithms. The source code for each of the algorithms was downloaded

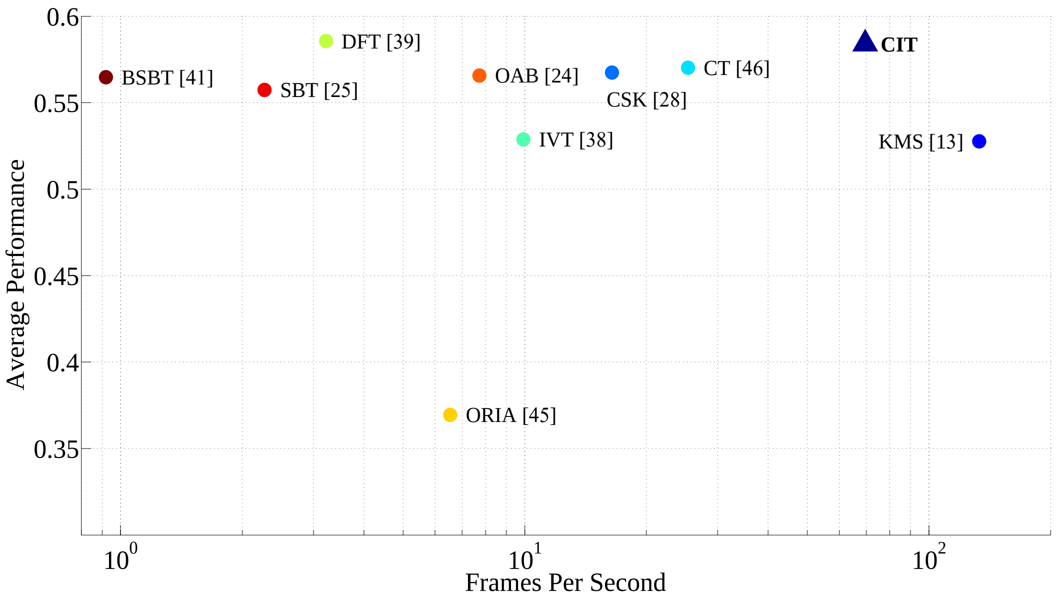


Figure 3.6: The average performance and frame rate for each tracking algorithm using the Buffy dataset. The average performance is the same as the value in the top plot of Fig 3.5. The results indicate that our method, CIT, is 10% more accurate and nearly as fast as the fastest of the competing algorithms. It is also as accurate but 20x faster than the most accurate of the competing algorithms.

from the original author's website. Minor modifications were then made so that each tracker would operate in our multi-target, automatic initialisation framework (as opposed to single target, human initialised frameworks that these algorithms were originally tested on). If an algorithm had parameters to set then those that were recommended by the original authors were used.

For all of the experiments we use the multi-scale ACF detector of Dollar *et al.* [16, 19] for the category detector. The code was downloaded from the website[2].

The category detector used for the Buffy dataset was trained using 882 faces from a separate dataset [13] which was also downloaded from the web. Due to the ground-truth for Buffy being noisy, an external dataset was used to train the face detector to ensure detection results were reasonable. The category detector used for the Caltech Pedestrians dataset was trained using the recommended training sets S0-S4 (refer to Sec.3.5).

Each category detector was then calibrated to operate at a specific recall by using a validation set. Episode 1 was used for Buffy and set S5 for Caltech Pedestrians. By operating the category detectors at different recall rates different targets are identified

[2] http://vision.ucsd.edu/~pdollar/toolbox/doc/

thus having an impact on the initialisation and update stages of tracking. We only include the results for a category detector recall of 0.5. Refer to the supplementary material for other values.

The validation sets were also used to calibrate each of the tracking methods. Each of the tracking methods have a confidence score associated with their predictions; in our regime, individuals enter and exit the scene so it is important to stop tracking a target when the confidence of the tracker is below a certain threshold otherwise the trackers will update their target model with background. To choose this threshold we evaluated each algorithm on the validation set for each dataset and selected the threshold that gave the best average performance. For the CIT tracker we selected the value of β (refer to Eqn. 3.3) during validation rather than the confidence threshold which was set to zero.

Each tracking algorithm was then evaluated on the test set for Buffy (episodes 2-5) and for Caltech Pedestrians (sets S6-S10) with trackers being initialised by non-coincident category detections (refer to Fig. 3.2). The resulting trajectories were used to compute precision, recall and continuity for each algorithm. The results for both datasets are in Figure 3.5, which includes the average performance, which is the mean of the precision, recall and continuity. The precision calculation ignores trajectories that have been initialised by false category detections.

The speed at which each of these algorithms operate at is also important. To compute the average frame rate of a tracking algorithm we need to decouple the computation time due to the tracker and the computation time due to the category detector. The time it takes for the category detector to run without tracking is subtracted from the time it takes a tracker to run. The results are in Figure 3.6. We only include the results for Buffy since the results on Caltech Pedestrians (in the supplementary material) are similar. The average frame rates were computed using the first 10,000 frames of episode 3 of Buffy on an Intel i5, 3.20 GHz machine.

Qualitative examples of how our tracking method works on both Buffy and Caltech Pedestrians can be found in Figure 3.1. The sequences give an indication of how tracking of a target is successful despite occlusion and changes in pose.

3.8 Discussion and Conclusions

We have presented a novel tracking method which is designed to track objects belonging to a specific category. The method makes use of a category detector to identify target objects to track and of an individual-specific detector to track the target in subsequent video frames. The individual-specific detector is trained on-the-fly at almost no extra cost, making it possible for the tracker to operate in real-time. The well-known problem of drift is addressed by updating the individual-specific detector only when there are coincident category detections.

We compare the performance of our scheme to 9 state-of-the-art trackers and find that it is as accurate as the most accurate competitor, but 20x faster. It is only slightly slower than the fastest competitor, but 10% more accurate.

In order to carry out our benchmark comparison we developed a methodology based on considering four metrics: precision, recall, fragmentation and computational cost. Our experiments were carried out on two large (hundreds of thousands of detections), challenging and heterogeneous datasets of faces and pedestrians; we observe identical rankings of the various algorithms on the two datasets, which gives us the confidence that our findings are general and may be expected to carry over to a variety of datasets and tasks. We believe that our benchmark surpasses, both in method and set size, any such comparative evaluation in the literature.

Acknowledgements

This work is funded by the ARO-JPL NASA Stennis NAS7.03001 grant and the ONR MURI Grant N00014-10-1-0933.

3.9 Appendix: Further Experiments

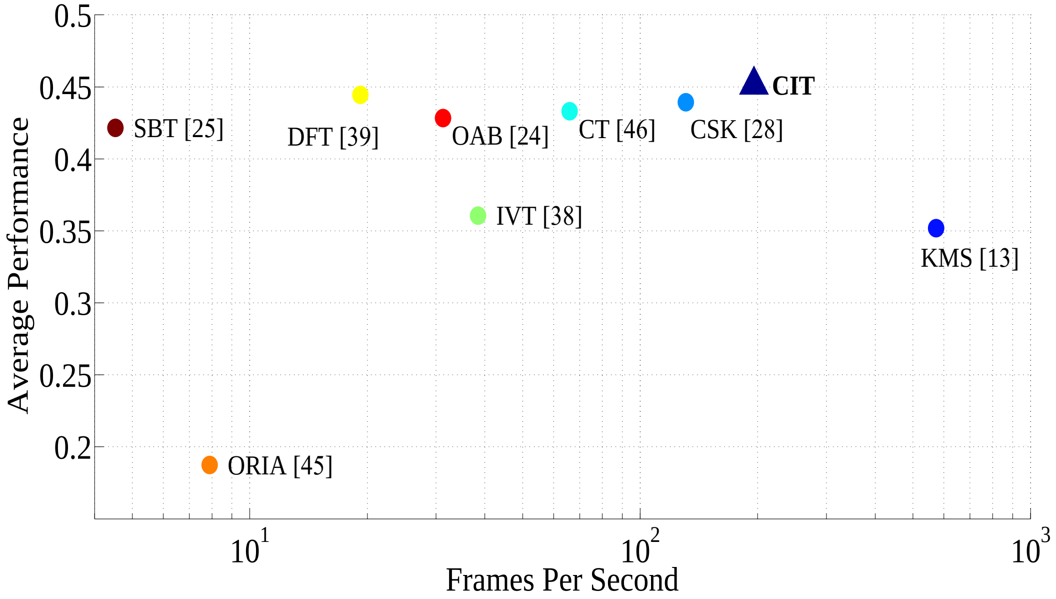


Figure 3.7: The average performance and frame rate for each tracking algorithm using the Caltech Pedestrians dataset. The average performance is the same as the value in the bottom plot of Figure 3.5. The results indicate that our method, CIT is the most accurate and slightly faster than the next most accurate. The relative performance and frame rate of the tracking methods are similar to those as reported on the Buffy dataset which can be found in Figure 3.6. However, the absolute frame rate has increased significantly between datasets due to Caltech Pedestrians having a lower resolution than Buffy.

In the supplementary material we report the results of some additional experiments. The most important findings are reported in the manuscript. In Figure 3.7 we report the average performance and frame rate of each tracking algorithm on the Caltech Pedestrians Dataset. In Figure 3.8 we report the tracking performance of each tracking algorithm for a category detector operating at different recall rates. The rankings of the algorithms across the different datasets and across the different category detector recall rates gives us confidence that our findings are general.

In Figure 3.9 we report results by evaluating the tracking algorithms on every 30th frame of the datasets rather than on every frame as in the previous analysis. This experiment demonstrates how well trackers cope with large changes in appearance and in position. CIT is not the best method in this case for some settings; however, it is still 10–90x faster than the methods that do have a better average performance.

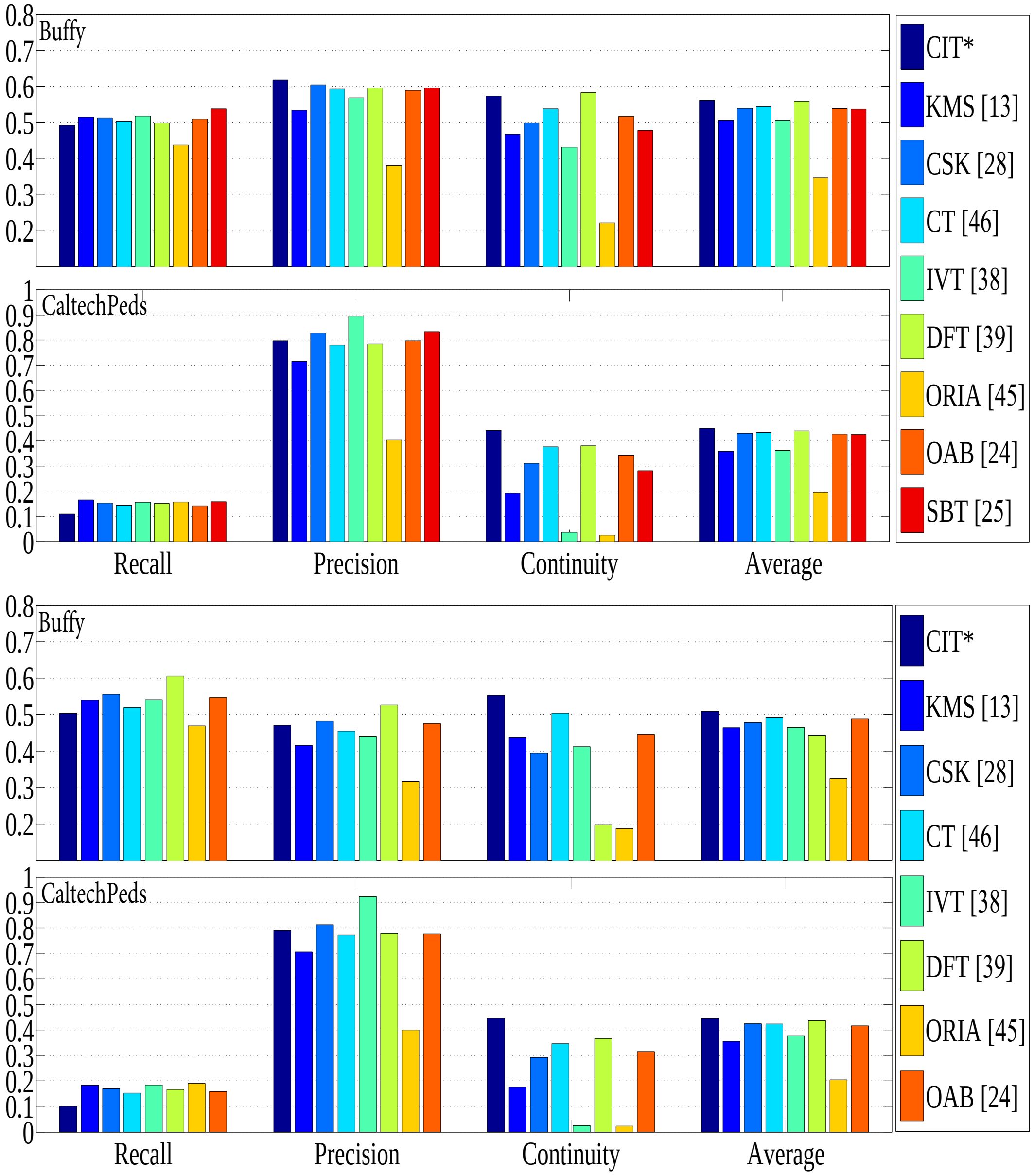


Figure 3.8: Tracking Performance on Buffy and Caltech Pedestrians for a category detector operating at a recall of (top) 0.6 [Buffy], 0.7 [CaltechPeds] and (bottom) 0.7 [Buffy], 0.9 [CaltechPeds]. The results indicate that our method CIT is one of the best performers, however, most of the methods have similar performance results. The relative performance of the trackers is roughly equivalent across the two datasets as well as across the different category detector recall rates. (BSBT and SBT were omitted because they operate too slowly; at higher recall rates more objects are identified to track and so the computational requirements increase.)

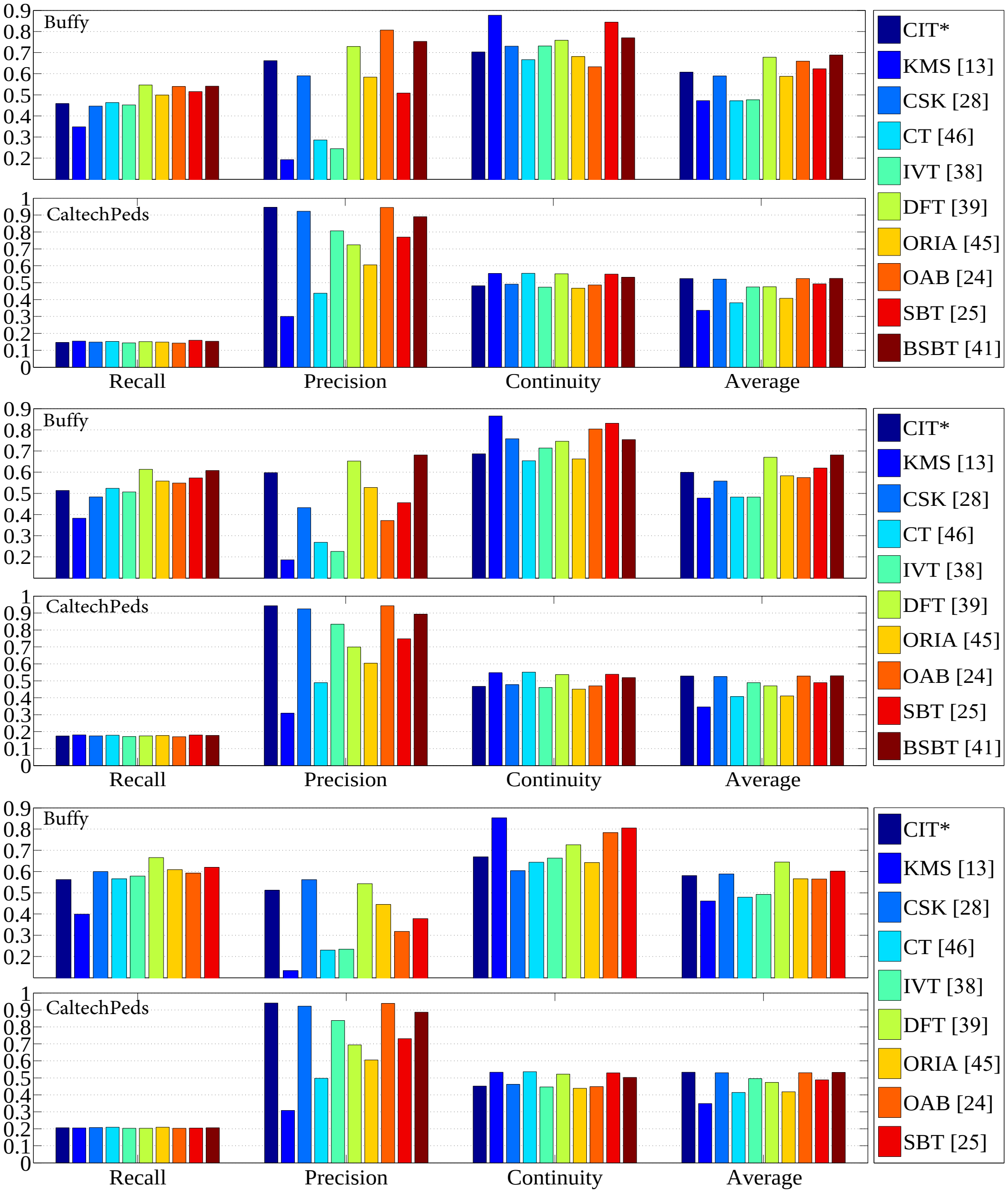


Figure 3.9: Tracking Performance with a 30 frame jump on Buffy and Caltech Pedestrians for a category detector operating at a recall of (top) 0.5 [Buffy], 0.5 [CaltechPeds] (middle) 0.6 [Buffy], 0.7 [CaltechPeds] and (bottom) 0.7 [Buffy], 0.9 [CaltechPeds]. Instead of evaluating the trackers on every single frame in the datasets, they are evaluated on every 30th frame. This gives an indication of how well trackers can handle large appearance changes and large changes in position from one frame to the next. The performance of the KMS tracker suffers because the target can have large changes in position from frame to frame. CIT, CSK, DFT, OAB, SBT and BSBT are all able to handle large changes reasonably well. CIT is outperformed by DFT, OAB, SBT and BSBT on Buffy for a category detector recall of 0.5; however, these are the slowest tracking methods.

References

[1] A. Adam, E. Rivlin, and I. Shimshoni. “Robust Fragments-based Tracking using the Integral Histogram”. In: *CVPR*. 2006. DOI: 10.1109/CVPR.2006.256.

[2] D. Hall and P. Perona. “From Categories to Individuals in Real Time — A Unified Boosting Approach”. In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2014. DOI: 10.1109/cvpr.2014.30.

[3] A. Andriyenko and K. Schindler. “Globally Optimal Multi-target Tracking on a Hexagonal Lattice”. In: *ECCV*. 2010. DOI: 10.1007/978-3-642-15549-9_34.

[4] S. Avidan. “Ensemble Tracking”. In: *PAMI* 29.2 (2007), pp. 431–435. DOI: 10.1109/CVPR.2005.144.

[5] B. Babenko, S. Belongie, and M.-H. Yang. “Visual Tracking with Online Multiple Instance Learning”. In: *CVPR*. 2009. DOI: 10.1109/CVPR.2009.5206737.

[6] C. Bao et al. “Real Time Robust L1 Tracker using Accelerated Proximal Gradient Approach”. In: *CVPR*. 2012. DOI: 10.1109/CVPR.2012.6247881.

[7] M. Bauml, M. Tapaswi, and R. Stiefelhagen. “Semi-supervised Learning with Constraints for Person Identification in Multimedia Data”. In: *CVPR*. 2013. DOI: 10.1109/CVPR.2013.462.

[8] R. Benenson et al. “Pedestrian Detection at 100 Frames per Second”. In: *CVPR*. 2012. DOI: 10.1109/CVPR.2012.6248017.

[9] J. Berclaz, F. Fleuret, and P. Fua. “Multiple Object Tracking using Flow Linear Programming”. In: *IEEE International Workshop on Performance Evaluation of Tracking and Surveillance*. 2009.

[10] J. Berclaz et al. “Multiple Object Tracking using K-Shortest Paths Optimization”. In: *PAMI* 33.9 (2011), pp. 1806–1819. DOI: 10.1109/TPAMI.2011.21.

[11] K. Bernardin and R. Stiefelhagen. “Evaluating Multiple Object Tracking Performance: The CLEAR MOT Metrics”. In: *EURASIP Journal on Image and Video Processing* 2008.1 (May 2008), pp. 1–10. DOI: 10.1155/2008/246309.

[12] M. J. Black and A. D. Jepson. “EigenTracking: Robust Matching and Tracking of Articulated Objects Using a View-Based Representation”. In: *International Journal of Computer Vision* 26.1 (1996), pp. 63–84. DOI: 10.1023/A:1007939232436.

[13] X. Burgos-Artizzu et al. “Merging Pose Estimates across Space and Time”. In: *BMVC*. 2013. DOI: 10.5244/C.27.58.

[14] D. Comaniciu, V. Ramesh, and P. Meer. "Kernel-based Object Tracking". In: *PAMI* 25.5 (2003). Ed. by V. Ramesh and P. Meer, pp. 564–577. DOI: 10.1109/TPAMI.2003.1195991.

[15] N. Dalal and B. Triggs. "Histograms of Oriented Gradients for Human Detection". In: *CVPR*. 2005. DOI: 10.1109/CVPR.2005.177.

[16] P. Dollar, S. Belongie, and P. Perona. "The Fastest Pedestrian Detector in the West". In: *BMVC*. 2010. DOI: 10.5244/C.24.68.

[17] P. Dollar et al. "Pedestrian Detection: A Benchmark". In: *CVPR*. 2009. DOI: 10.1109/CVPR.2009.5206631.

[18] P. Dollár et al. "Pedestrian Detection: An Evaluation of the State of the Art". In: *PAMI* 34.4 (Apr. 2012), pp. 743–61. DOI: 10.1109/TPAMI.2011.155.

[19] P. Dollar et al. "Fast Feature Pyramids for Object Detection". In: *PAMI* 36 (2014), pp. 1532–1545. DOI: 10.1109/TPAMI.2014.2300479.

[20] P. Dollár et al. "Integral Channel Features". In: *BMVC*. 2009. DOI: 10.5244/C.23.91.

[21] M. R. Everingham, J. Sivic, and a. Zisserman. "Hello! My name is... Buffy" – Automatic Naming of Characters in TV Video". In: *BMVC*. 2006. DOI: 10.5244/C.20.92.

[22] M. Everingham et al. *The PASCAL Visual Object Classes Challenge 2009 (VOC) Results*. 2009.

[23] Y. Freund and R. E. Schapire. "A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting". In: *Journal of Computer and System Sciences* 55.1 (1997), pp. 119–139. DOI: 10.1006/jcss.1997.1504.

[24] K. Fukunaga and L. Hostetler. "The Estimation of the Gradient of a Density Function, with Applications in Pattern Recognition". In: *IEEE Transactions on Information Theory* 21 (1975). DOI: 10.1109/TIT.1975.1055330.

[25] H. Grabner, M. Grabner, and H. Bischof. "Real-time Tracking via On-line Boosting". In: *BMVC* (2006). DOI: 10.5244/C.20.6.

[26] H. Grabner, C. Leistner, and H. Bischof. "Semi-supervised On-line Boosting for Robust Tracking". In: *ECCV*. 2008. DOI: 10.1007/978-3-540-88682-2.

[27] D. Gray and H. Tao. "Viewpoint Invariant Pedestrian Recognition with an Ensemble of Localized Features". In: *ECCV*. 2008. DOI: 10.1007/978-3-540-88682-2_21.

[28] J. F. Henriques et al. "Exploiting the Circulant Structure of Tracking-by-Detection with Kernels". In: *ECCV*. 2012. DOI: 10.1007/978-3-642-33765-9_50.

[29] G. B. Huang et al. “Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments”. In: *University of Massachusetts Amherst Technical Report 07* 49 (2007), pp. 1–11. DOI: 10.1.1.122.8268.

[30] H. Jiang, S. Fels, and J. J. Little. “A Linear Programming Approach for Multiple Object Tracking”. In: *CVPR*. 2007. DOI: 10.1109/CVPR.2007.383180.

[31] Z. Kalal, K. Mikolajczyk, and J. Matas. “Tracking-Learning-Detection”. In: *PAMI* 34.7 (July 2012), pp. 1409–1422. DOI: 10.1109/TPAMI.2011.239.

[32] Y. L. Y. Li, C. H. C. Huang, and R. Nevatia. “Learning to Associate: HybridBoosted Multi-target Tracker for Crowded Scene”. In: *CVPR* (2009). DOI: 10.1109/CVPR.2009.5206735.

[33] Y. Ma, Q. Yu, and I. Cohen. “Target Tracking with Incomplete Detection”. In: *Computer Vision and Image Understanding* 113.4 (2009), pp. 580–587. DOI: 10.1016/j.cviu.2009.01.002.

[34] R. Nevatia. “Global Data Association for Multi-object Tracking using Network Flows”. In: *CVPR*. 2008. DOI: 10.1109/CVPR.2008.4587584.

[35] “Online Robust Image Alignment via Iterative Convex Optimization”. In: *CVPR*. 2012. DOI: 10.1109/CVPR.2012.6247878.

[36] S. Oron et al. “Locally Orderless Tracking”. In: 2012. DOI: 10.1109/CVPR.2012.6247895.

[37] S. Pellegrini, A. Ess, and L. V. Gool. “Improving Data Association by Joint Modeling of Pedestrian Trajectories and Groupings”. In: *ECCV*. 2010. DOI: 10.1007/978-3-642-15549-9_33.

[38] H. Pirsiavash, D. Ramanan, and C. C. Fowlkes. “Globally-optimal Greedy Algorithms for Tracking a Variable Number of Objects”. In: *CVPR*. 2011. DOI: 10.1109/CVPR.2011.5995604.

[39] D. A. Ross et al. “Incremental Learning for Robust Visual Tracking”. In: *International Journal of Computer Vision* 77 (2007), pp. 125–141. DOI: 10.1007/s11263-007-0075-7.

[40] L. Sevilla-Lara and E. Learned-Miller. “Distribution Fields for Tracking”. In: *CVPR*. 2012. DOI: 10.1109/CVPR.2012.6247891.

[41] J. Sivic, M. Everingham, and A. Zisserman. “Who are you? - Learning Person Specific Classifiers from Video”. In: *CVPR*. 2009. DOI: 10.1109/CVPR.2009.5206513.

[42] S. Stalder, H. Grabner, and L. V. Gool. “Beyond Semi-supervised Tracking: Tracking should be as Simple as Detection, but not Simpler than Recognition”. In: *ICCV Workshops*. 2009. DOI: 10.1109/ICCVW.2009.5457445.

[43] P. Viola and M. Jones. “Rapid Object Detection using a Boosted Cascade of Simple Features”. In: *CVPR*. 2001. doi: 10.1109/CVPR.2001.990517.

[44] P. Viola and M. J. Jones. “Robust Real-Time Face Detection”. In: *Int. J. Comput. Vision* 57.2 (2004), pp. 137–154. doi: 10.1023/B:VISI.0000013087.49260.fb.

[45] Y. Wu, J. Lim, and M.-H. Yang. “Online Object Tracking: A Benchmark”. In: *CVPR*. 2013. doi: 10.1109/CVPR.2013.312.

[46] K. Zhang, L. Zhang, and M.-h. Yang. “Real-Time Compressive Tracking”. In: *ECCV*. 2012. doi: 10.1007/978-3-642-33712-3_62.

Chapter 4

FINE-GRAINED CLASSIFICATION

The contents of this chapter are adapted from the peer-reviewed publication "Fine-Grained Classification of Pedestrians in Video: Benchmark and State of the Art" by D. Hall and P. Perona, appearing at the Conference on Computer Vision and Pattern Recognition (CVPR) 2015[1]. The study of how temporal information can be utilised for the fine-grained classification of people, as well the experiments using the unified model, is unpublished work.

4.1 Abstract

A video dataset that is designed to study fine-grained classification of pedestrians is introduced. Pedestrians were recorded "in-the-wild" from a moving vehicle. Annotations include bounding boxes, tracks, 14 keypoints with occlusion information and the fine-grained categories of age (5 classes), sex (2 classes), weight (3 classes) and clothing style (4 classes). There are a total of 27,454 bounding box and pose labels across 4222 tracks. This dataset is designed to train and test algorithms for fine-grained classification of people. A unified model is introduced that takes a single image as input and outputs the class distributions for each of the four fine-grained categories in the dataset. A study of how temporal information can be utilised for the fine-grained classification of people is also conducted. Results show that the class average accuracy when combining information from a sequence of images of an individual and then predicting the label is 3.5-7.1% better than independently predicting the label of each image, when severely under-represented classes are ignored. The dataset is also useful for benchmarking tracking, detection and pose estimation of pedestrians.

4.2 Introduction

People are an important component of a machine's environment. Detecting, tracking, and recognising people, interpreting their behaviour and interacting with them is a valuable capability for machines. Using vision to estimate human attributes such as: age, sex, activity, social status, health, pose and motion patterns is useful for interpreting and predicting behaviour. This motivates our interest in fine-grained

[1] Project Website: http://vision.caltech.edu/~dhall/projects/CRP/

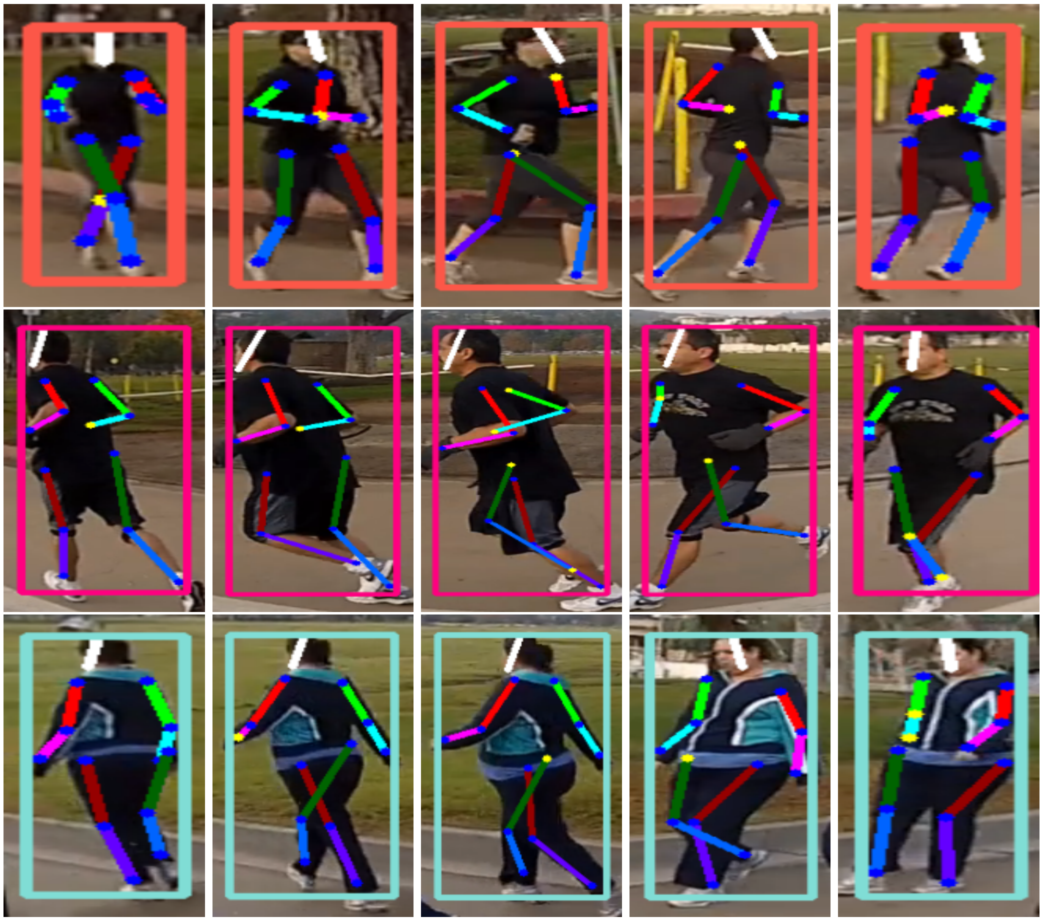


Figure 4.1: Three examples from the CRP dataset. Annotations include a bounding box, tracks, parts, occlusion, sex, age, weight and clothing style.

classification of people.

Visual classification involves recognising basic categories (e.g. 'birds' vs. 'chairs') and fine-grained categories, also called subcategories (e.g. 'barn swallow' vs. 'marten') [7]. Since subcategories are similar in appearance subtle differences are often crucial. This is in contrast to basic classification where categories are visually distinct and therefore broad statistics of the image are often sufficient.

While research in broad classification is well-supported by large datasets comprising thousands of categories [15, 33], fine-grained classification has so far been explored in a small number of domains including: animal breeds and species [8, 29, 53], plant species [43], objects [50, 36] and, what will be the focus of this work, people [9, 13, 6, 56]. The availability of good-quality, large annotated datasets covering as many domains as possible is crucial for progress in fine-grained classification.

Prior work on the fine-grained classification[2] of people has typically focused on *faces* with subcategories including: identity, age, sex, clothing type, facial hair and skin colour [14, 20, 40, 48, 3, 31].

Fine-grained classification using the entire human body is still a relatively unexplored area. The current benchmark using images is "The Attributes of People Dataset" [6] which was introduced in 2011. It includes nine subcategories and has large variations in viewpoint and occlusion. This dataset has been useful for researchers working on human attribute recognition [6, 56] but is limited by a) its size, particularly when training deep networks [56], b) only bounding box annotations are provided and c) all subcategories are binary.

There are also a number of video-based benchmark datasets [47, 24, 37] that are geared towards gait recognition. These datasets are limited by a) the raw video footage is difficult to obtain with only silhouettes being readily available, b) subjects are cooperative (they know they are being filmed), c) the background is static and uncluttered, d) viewpoints are all profile and e) subcategories are limited to identity and sex.

In this work, the first contribution we make is the introduction of a public video dataset—Caltech Roadside Pedestrians (CRP)—to further advance the state-of-the-art in fine-grained classification of people using the entire human body. Its novel and distinctive features are:

1. Size (27,454 bounding box and pose labels) – making it suitable for training deep-networks.
2. Natural behaviour – subjects are unaware, and behave naturally.
3. Viewpoint – Pedestrians are viewed from front, profile, back and everything in between.
4. Moving camera – More general and challenging than surveillance video with static background.
5. Realism – There is a variety of outdoor background and lighting conditions; examples can be found in Figure 4.25.
6. Multi-class subcategories – age, clothing style and body shape.

[2] In the literature, fine-grained classification is more commonly referred to as attribute recognition in the human domain.

7. Detailed annotation – bounding boxes, tracks and 14 pose keypoints with occlusion information; examples can be found in Figure 4.1. Each bounding box is also labelled with the fine-grained categories of age (5 classes), sex (2 classes), weight (3 classes) and clothing type (4 classes).

8. Availability – All videos and annotations are publicly available

The second contribution is the introduction of a unified model for the fine-grained classification of people using single images. This model is a neural network that takes a single image as input and outputs the class distributions for each of the four fine-grained categories in the CRP dataset.

Since this is a video dataset, **the final contribution is a study of how temporal information can be utilised for the fine-grained classification of people.** Using sequences of images of an individual, we explore different strategies for combining that information so that a consistent fine-grained classification prediction is obtained. To the best of our knowledge this is the first time that the fine-grained classification of pedestrians in video has been studied.

4.3 Related work

Existing fine-grained datasets on birds [8, 52, 5], dogs [29, 34, 45], cats [45], butterflies [53], flies [38], leaves [32], flowers [43, 42], aircraft [36] and cars [50] cover a single subcategory (usually species, breed or type) but have hundreds of classes.

Fine-grained classification of people, however, began with a focus on the single *binary* subcategory of sex. Using neural networks, SEXNET [20] and EMPATH [14] were the first efforts to classify sex from faces; methods using support vector machines [40] and boosting [48, 3] soon followed. Work on classifying age and race from faces can also be found in the literature, with in-depth surveys available [19, 18]. The first attempt at collecting a face dataset with multiple, multi-class subcategories was FaceTracer [31]. It captured seven subcategories relevant to people, these included sex, age (4 classes) and race (3 classes). Each subcategory had between 1000-4000 examples.

In low resolution situations and particularly surveillance settings, faces are not suitable for fine-grained classification. This has led to work that looks at using the entire body, which presents additional cues such as clothing, body shape and motion patterns. Cao [9] took the existing MIT pedestrian dataset [44] and manually annotated it with sex labels; this was repeated by [13] who also labelled the VIPeR [21] dataset. It wasn't until "The Attributes of People Dataset" [6] was released in 2011, that a full-body dataset, with more than a single subcategory, was publicly available. This dataset has 9 binary subcategories across 8035 images, with large variations in viewpoint and occlusion. Bounding box annotations are also provided. The dataset has a high resolution with an average bounding box size of 532 x 298 pixels and is the current benchmark. More recently, the "Attributes 25K Dataset" [56] was collected; it is a large dataset with 24,963 examples and the same subcategories as [6]; however, it is not publicly available.

The gait recognition community utilise the temporal information available in video for fine-grained classification, with a particular focus on identity. The task here is to infer the identity of someone from the way they walk. State-of-the-art methods for gait recognition extract a *sequence* of silhouettes of a particular person. A Gait Energy Image [22] or a Gait Entropy Image [4] is then computed from the sequence of silhouettes once an estimate of the gait period is determined. These features are then used for classification.

There are a number of video datasets available, the first being the USF HumanID Dataset [47]. It contains 1870 sequences of 122 individuals. It is collected outdoors with a static background. Participants are cooperating subjects, aware of being filmed, who are asked to walk a predefined elliptical path, with only the back portion used ensuring the viewpoint is always profile. Silhouettes are available for immediate download while the entire video collection can take up to 3 months to obtain. A more recent example is the Large Population Dataset [24]. It contains 4016 subjects who each occur in 2 sequences. It is collected indoors with a green screen background. Participants walk a predefined path, however, the viewpoint varies from nearly-frontal to profile. Sex and age labels were also collected, however these have not yet been released. Only silhouettes are available for download but only after authorisation is granted by the authors.

While utilising temporal information for fine-grained classification is relatively unexplored, it is a well studied field in the person re-identification [39, 35] and action recognition [2, 17, 23, 25, 28, 41, 49] domains.

For activity recognition Donahue *et al*. [17] extracts features from each frame of a video sequence using a convolutional neural network. These features are then passed through an RNN which, at each time step, outputs the predicted class distribution for the activity recognition task. These class distributions are then averaged to produce

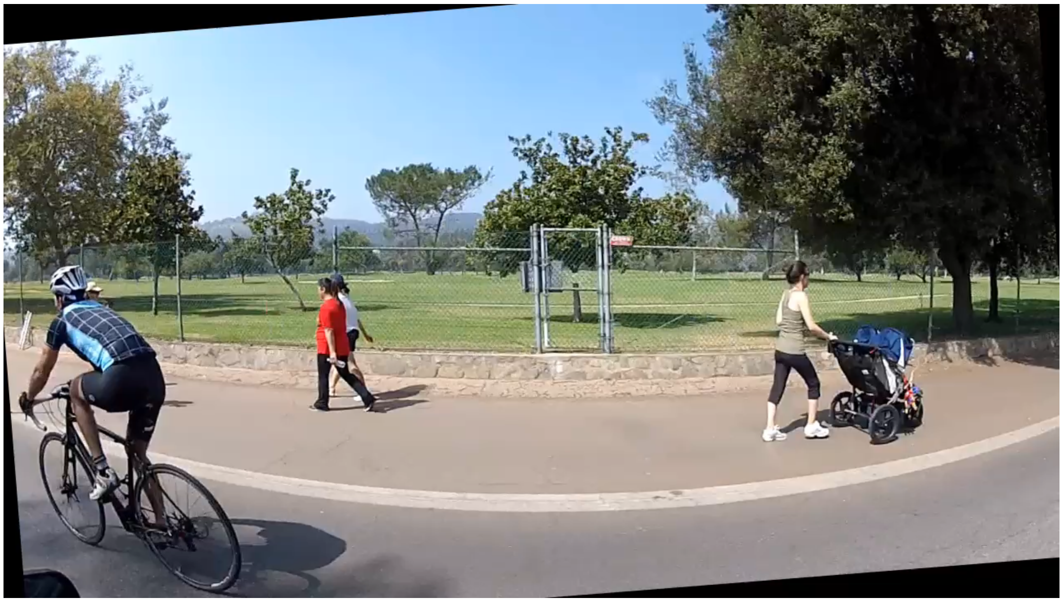


Figure 4.2: An example frame from the Caltech Roadside Pedestrian Dataset. In digital versions of this work, clicking on the image will redirect you to YouTube to view one of the videos in the CRP dataset.

the final prediction. Singh *et al*. [49] have a similar approach but use a bi-directional LSTM. McLaughlin *et al*. [39] also have a similar methodology but apply it to the person reidentification task. Both Ng *et al*. [41] and Karpathy *et al*. [28] explore multiple temporal feature pooling architectures when experimenting with activity recognition. They both look at what stage of a deep neural network temporal features should be pooled, whether it be at the input image stage, at the end of the network or somewhere in between. They also consider whether features should be fused using max or average pooling or whether the fusion should be performed by a fully connected layer.

4.4 Dataset Collection

In this section we describe in detail, the method in which the videos of the dataset were collected and annotated. Due to the large number of annotations required it was important to develop an efficient and cost effective pipeline. For this reason, crowdsourcing, using workers from Amazon's Mechanical Turk (MTURK) was used for all of the annotation tasks.

Video Collection

This dataset contains 7 videos. Each video is captured by mounting a rightwards-pointing, GoPro Hero3 camera to the roof of a car. The car then completes three

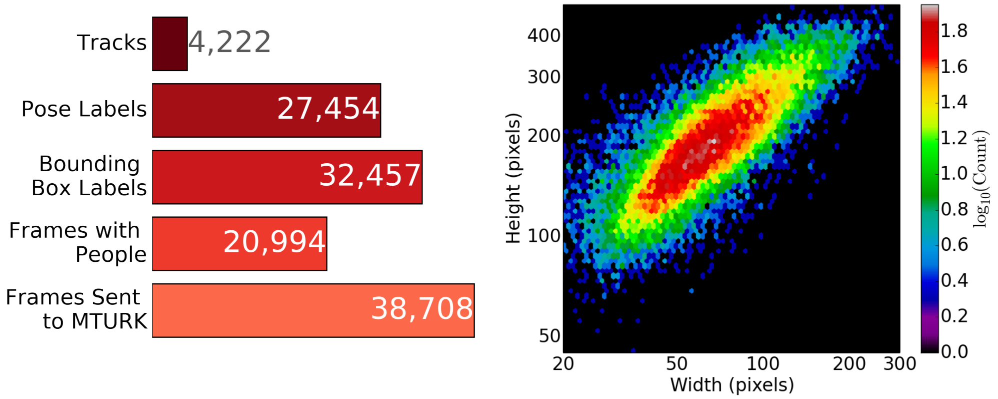


Figure 4.3: (Left) Dataset Statistics. **(Right)** A histogram of the height and width of the bounding boxes in the dataset. The mean bounding box size is 71 pixels wide by 201 pixels high. The resolution is twice as large as Caltech Pedestrians [16] but 2.5 times smaller than the "Attributes of People Dataset", which is currently used for fine-grained classification [6].

laps of a ring road within a park where there are many walkers and joggers. The videos were shot using a wide-angle mode, at a resolution of 1280x720 pixels, and a frame rate of 30 fps. Each video has on average, 37,000 frames, for a total of 261,645 frames in the entire dataset. Each video was recorded at 8AM on different days of the week, over a 9 month period.

Bounding Box Annotation

For each video, the first task was to annotate all of the pedestrians with bounding boxes. To make this a cost-effective task, a coarse-to-fine approach was used. Every 10th frame was sent to MTURK where three workers were instructed to draw a bounding box around every pedestrian in the image. The bounding boxes from each worker were then combined into a single set of bounding box labels for each frame using clustering. For this stage, a total of 26,168 frames were sent for annotation. Figure 4.21 contains an example of the interface.

A further set of frames were sent for annotation so that every 5th frame of the video would be labelled. To avoid sending empty frames, the results from the coarse labelling attempt were used. For every frame x that had a set of bounding box labels (the image actually contained pedestrians), two frames were sent to MTURK for labelling, frames $x + 5$ and $x - 5$. These frames were again annotated by three workers and a single set of bounding box labels were generated as before. For this stage, 12,540 frames were sent for annotation. A total of 32,457 bounding box annotations were collected.

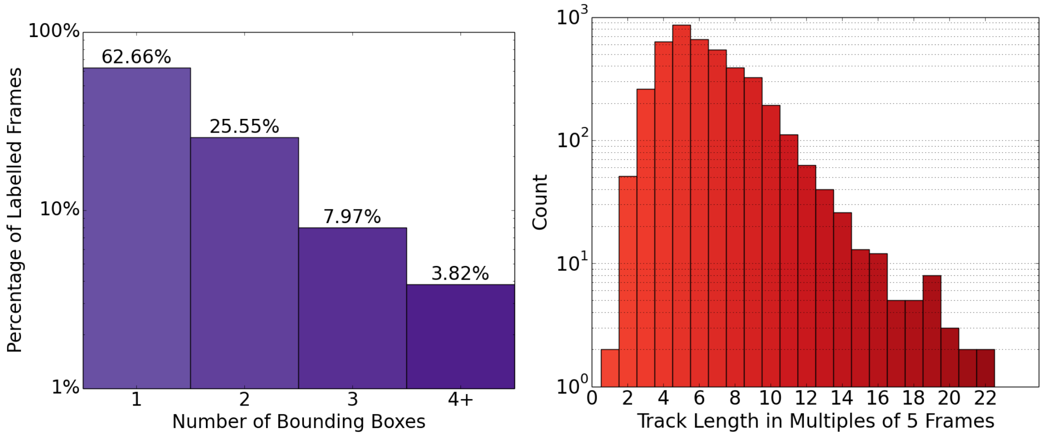


Figure 4.4: (Left) A histogram of the number of bounding boxes in each of the labelled frames in the dataset. 63% of frames have only a single person in them, the remainder have two or more. **(Right) A histogram of the track length.** Since only every fifth frame is labelled, each pedestrian takes an average of 32.5 frames or 1.1 seconds to move through the field of view of the camera.

Track Annotation

The next task was to create tracks (the time trajectory of an individual) from the bounding boxes. To do this, a worker was given a cropped image of a person from frame x (obtained using the bounding box labels). They were then instructed to make a selection from the set of cropped people from frame $x + 5$ that matched the original image. There was also an option to select that there was no match. Every person with a bounding box over 100 pixels in height was labelled by three workers. The workers' annotations were combined using a majority vote. If there was disagreement between all three workers, the bounding box was assigned a no-match label. Tracks were formed by chaining together the bounding boxes until a no-match label was encountered. Tracks were then verified by an expert annotator. Their task was to eliminate short tracking gaps and to correct any other mistakes. A total of 4,222 tracks were collected. An example of the interface can be found in Figure 4.22.

Pose Annotation

To represent the pose of a human body, 14 body parts were used as keypoints: top of the head, chin, the right and left shoulders, elbows, wrists, hips, knees, and ankles. These are the same parts used in existing datasets [27, 46]. To annotate the parts, workers were given a cropped image of a person (obtained using the bounding box labels but with some extra padding) and instructed to click on one of the 14 body

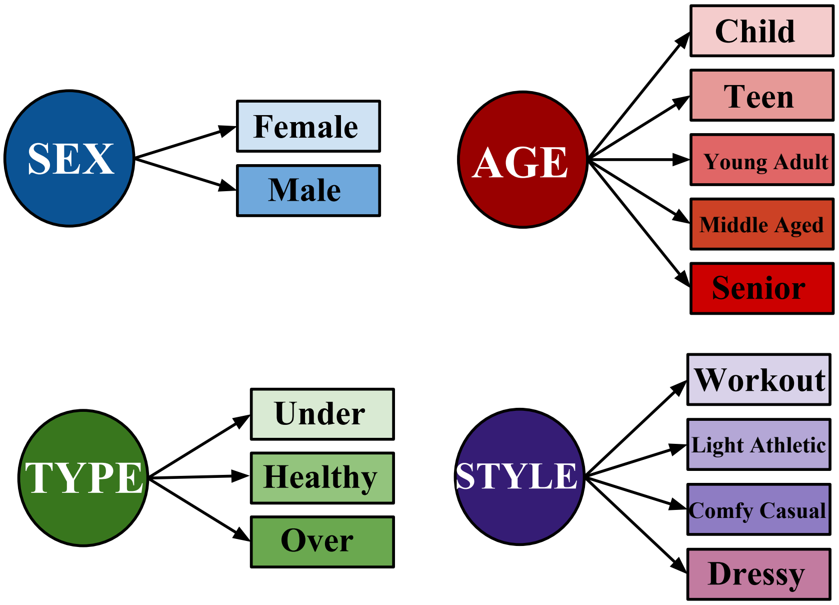
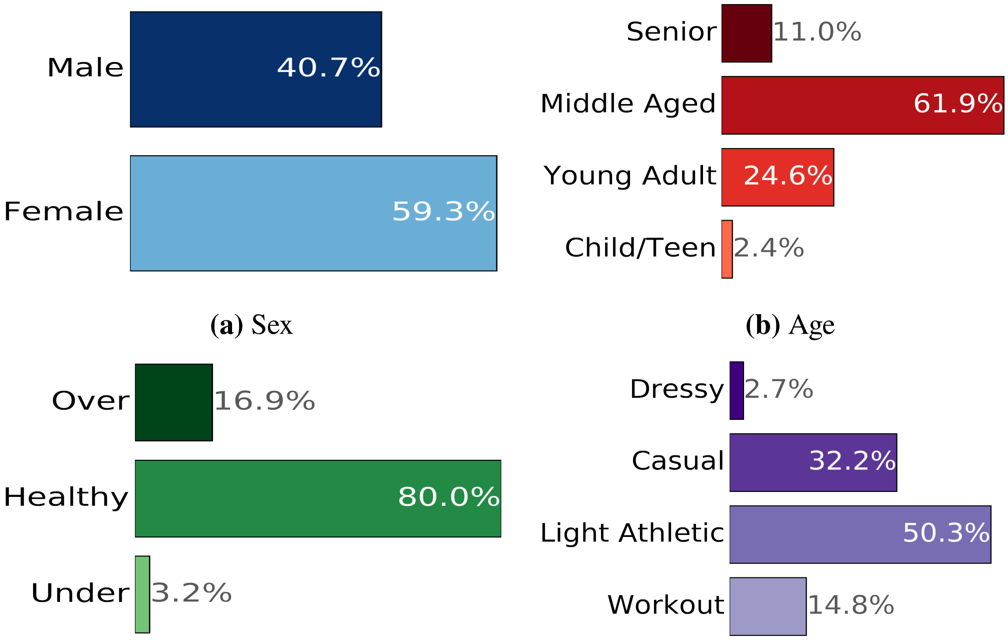


Figure 4.5: The possible class labels for each of the four subcategories: sex, age, weight and clothing style

parts. If the part was occluded in any way, the workers were asked to right click where they thought the part was located. It was also possible to indicate if a part was not in the image. Every person with a bounding box over 100 pixels in height was labelled by three workers for each of the 14 keypoints. The workers annotations were combined by taking the median of the labels. A total of 27,454 pose annotations with 14 keypoints and occlusion information were collected. The pose annotations were then used to refine the bounding box labels since the worker labelled bounding boxes were not always tight. The refined bounding box is the tightest box that covers the set of keypoints. An example of the interface can be found in Figure 4.23.

Fine-Grained Category Annotation

As mentioned in Section 4.2, using vision to estimate human attributes is useful for interpreting and predicting behaviour. In this dataset we look at four fine-grained categories of people: sex, age, weight and clothing style. The possible class labels for these subcategories are shown in Figure 4.5. While the classes for sex, age and weight were intuitive for us, those for clothing style were not. The four clothing style classes (workout, light athletic, comfortable casual and well-dressed (or dressy)) were chosen in consultation with a fashion expert. The 'workout class' groups people wearing spandex, singlet tops or no shirt at all. The 'light athletic' class includes people who are wearing yoga pants and tracksuits. The 'comfortable



(a) Sex

(b) Age

(c) Body Type

(d) Clothing Style

Figure 4.6: The percentage of labels for each class in the four fine-grained categories of the dataset. For all subcategories there is quite a large class imbalance. Very few children, teenagers, under weight or well-dressed people were seen. Given the setting, this is not surprising. The remaining classes have a reasonable number of labels.

casual' class contains people wearing shorts or items of clothing that would be typically worn in a casual setting. The 'well-dressed' class are of people with button-up or collared shirts and dresses. Examples of these classes can be found in Figure 4.25

To annotate the fine-grained categories, workers were given 4 examples of a person, sampled from one of the tracks. This allowed the worker to see the person from all possible viewpoints. They were then asked to select the best class label for one of the subcategories. For the clothing style task, workers were also shown examples of each class. The workers fine-grained labels were combined by taking a majority vote. If there was complete disagreement between workers a further 2 workers labelled the track and another majority vote was taken. An example of the interface can be found in Figure 4.24.

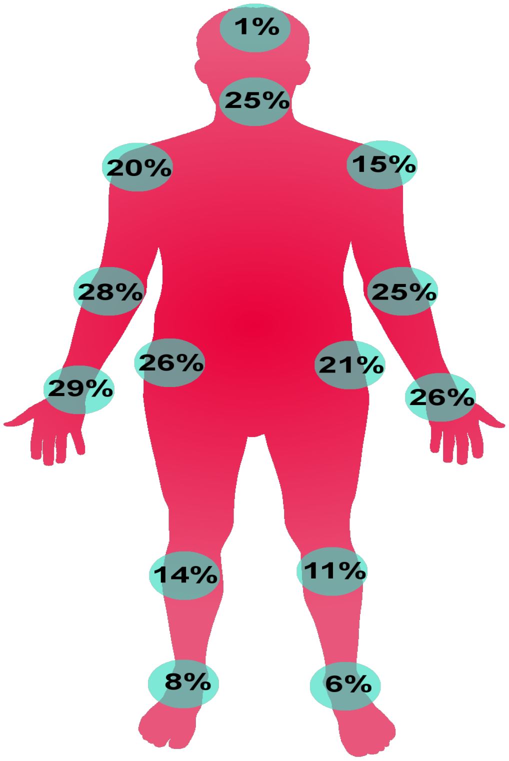


Figure 4.7: The percentage of labels that are occluded for each keypoint. The top of the head is rarely occluded, followed by the left and right ankles. The wrists, shoulders, elbows, hips and chin are all occluded around 25% of the time.

4.5 Dataset Analysis

In this section, we analyse labelling error and explore the properties of the dataset. A summary of the dataset's statistics can be found in Figure 4.3. The distribution of bounding box widths and heights, the number of people per frame, and the distribution of track lengths can be found in Figure 4.4. Occlusion statistics for pose keypoints and a breakdown of the the number of labels for each class in each of the four subcategories can be found in Figure 4.7.

Since the final bounding box estimates were derived from the pose labels and the tracks were verified by an oracle, the analysis of labelling error is focused on keypoints and fine-grained classes.

Pose Error

To estimate pose error, we take a keypoint from a particular sample (an image of a person). Since the location (x and y co-ordinates) of this keypoint was labelled by three different workers, the distance between each of the three locations can then be computed. The distance is normalised with respect to the height of the bounding

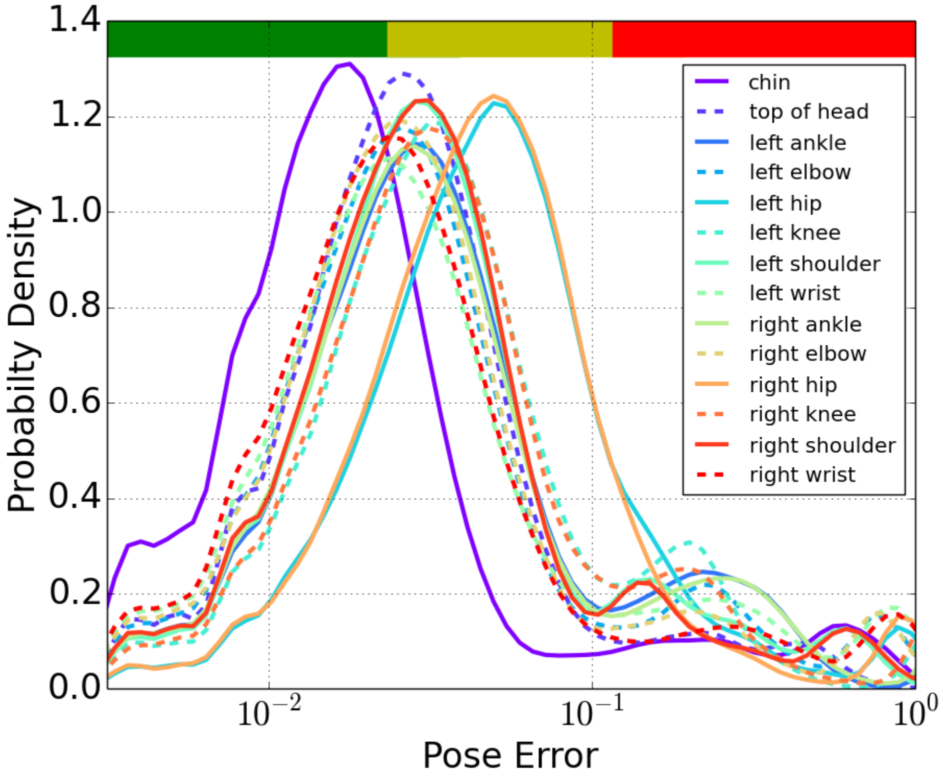


Figure 4.8: An estimate of keypoint location error as a fraction of body height. Given an image of a person, pose error is the distance between all of the workers keypoint locations, normalised by the height of the sample. The green band is where error is less than 3%, the yellow band between 3-15% and the red band is greater than 15% (See Section 4.5).

box for the sample. The distribution of these distances across all samples for each of the 14 keypoints can be found in Figure 4.8.

The results indicate that workers tended to agree the most about the location of the chin. This is expected since the chin is a sharp, well defined point on the face. The most disagreement occurred for the left shoulder and right hip. Both of these body parts are harder to localise since they are not sharp points.

The keypoints are classified into three error classes: excellent (the green band), where error is less than 3%, good (yellow band), where error is between 3-15% and poor (red band) where error is greater than 15%. The poor errors tend to occur when workers incorrectly exchange left and right labels.

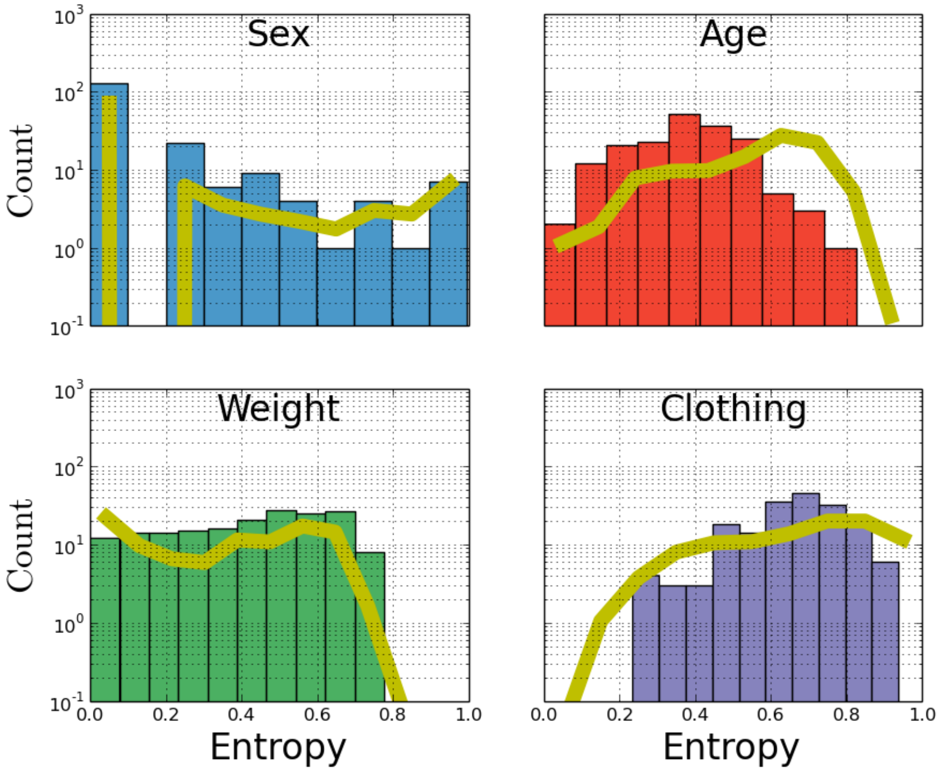


Figure 4.9: Estimates of the fine-grained labelling error for each subcategory. The histograms correspond to the entropy of 30 workers labelling 180 randomly selected tracks from the dataset. The yellow line corresponds to our statistical model as described in Section 4.5. The noise values in the model are 0.07, 0.15, 0.11 and 0.23 for sex, age, weight and clothing respectively. They need to be compared to the size of the bins, which depends on the number of classes in the subcategory. Bin sizes are equal to 0.5, 0.2, 0.33 and 0.25 respectively.

Fine-Grained Label Error

To estimate the error in the fine-grained labels, 180 tracks were randomly selected from the dataset and sent to MTURK to be labelled by 30 workers. The same process as outlined in Section 4.4 was followed except for the increase in annotators.

Given a sample track, for each subcategory, the empirical probability distribution of its classes was calculated from the 30 worker labels. The normalised entropy for the sample was then computed using this distribution. A histogram of entropies for each subcategory across the 180 examples can be found in Figure 4.9. Low entropies indicate high agreement amongst annotators.

The fine-grained label error may be modelled statistically. Each one of the attributes we considered may be thought as varying in one dimension; thus, we assume that the L class labels of a given attribute are the result of discretising a uniformly distributed

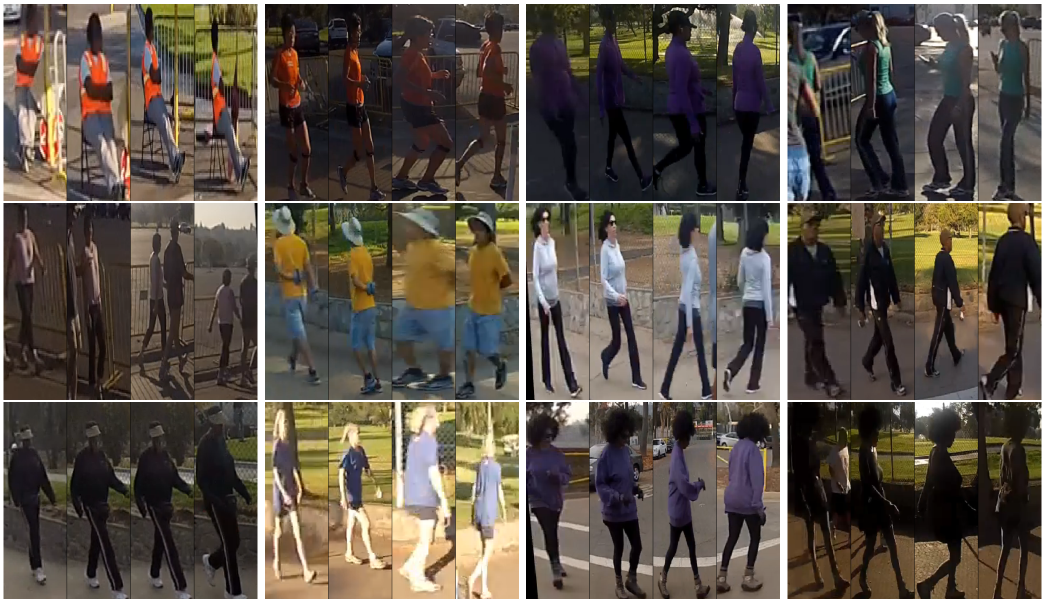


Figure 4.10: The most ambiguous cases to label for sex (far left), age (centre left), weight (centre right), and clothing style (far right), based on the entropy from the set of images labelled by 30 workers.

continuous variable (e.g. age ranges from 0 to 100 years, and it is binned into five age categories).

We assume that each annotator may estimate the underlying continuous variable, with the addition of some 'annotator noise' and produce a label by binning their continuous estimate. We model annotator noise as zero-mean with a free parameter σ, which we assume constant over the population of the annotators (a more sophisticated point of view may be found in [54]). In order to estimate σ we fit this model to the empirical labelling results for each of the four subcategories (for convenience we rescaled the range of the underlying continuous variable to $(0, 1)$). Results are shown in Figure 4.9. The noise values that best fit the data are 0.07, 0.15, 0.11 and 0.23 for sex, age, weight and clothing respectively. This has to be compared to the size of the bins, which depends on the number of classes and is thus equal to 0.5, 0.2, 0.33 and 0.25 for each subcategory respectively. This means that the annotators' estimates of sex are excellent, quite consistent for weight, and somewhat vague for age and clothing, as one might expect. Figure 4.10 contains examples of the three most ambiguous samples for each of the four subcategories.

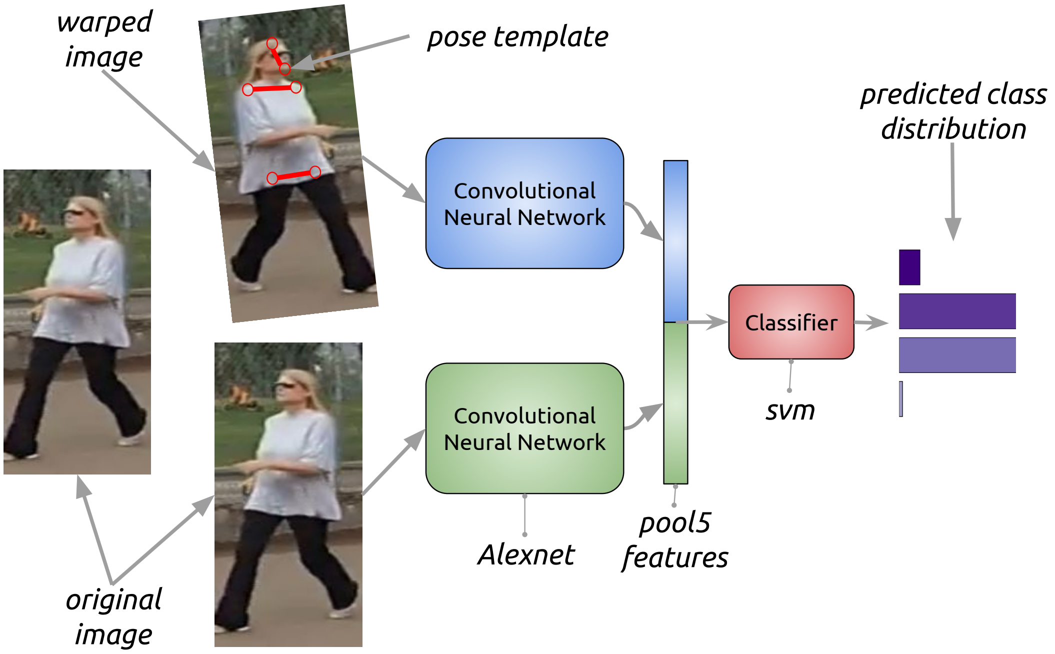

warped image
pose template
predicted class distribution
Convolutional Neural Network
Classifier
svm
Convolutional Neural Network
Alexnet
pool5 features
original image

Figure 4.11: Baseline Model. The baseline model used is the 'pose normalised deep convolutional nets' method of Branson *et al.* [7]. Given an image containing a person, it is warped using a similarity transform to fit a prototypical pose template that has been hand defined. This warped image and the original image are then fed through a convolutional neural network with an AlexNet [30] architecture, pre-trained on ImageNet [26]. Features are extracted from the 5th layer after max-pooling and then concatenated. Separate one-vs-all linear SVM's are then trained to classify each of the fine-grained categories.

4.6 Experiments: Fine-Grained Classification - Single Images

In this section we present two sets of fine-grained classification experiments. The first set considers only single images of people while the second set uses sequences of people. The dataset is split into a training/validation set containing four videos, with the remaining three videos forming the test set. Since each video was collected on a unique day, different images of the same person **do not** appear in both the training and testing sets. While it is conceivable that a person appears in different videos while wearing the same clothes and in the same lighting conditions, we consider this to be unlikely.

Baseline Model: Pose Normalised Deep Convolutional Nets

The baseline model we use is the 'pose normalised deep convolutional nets' method of Branson *et al.* [7]. It has state-of the-art performance on bird species classification and we believe that the method will generalise to the people dataset. This is the only

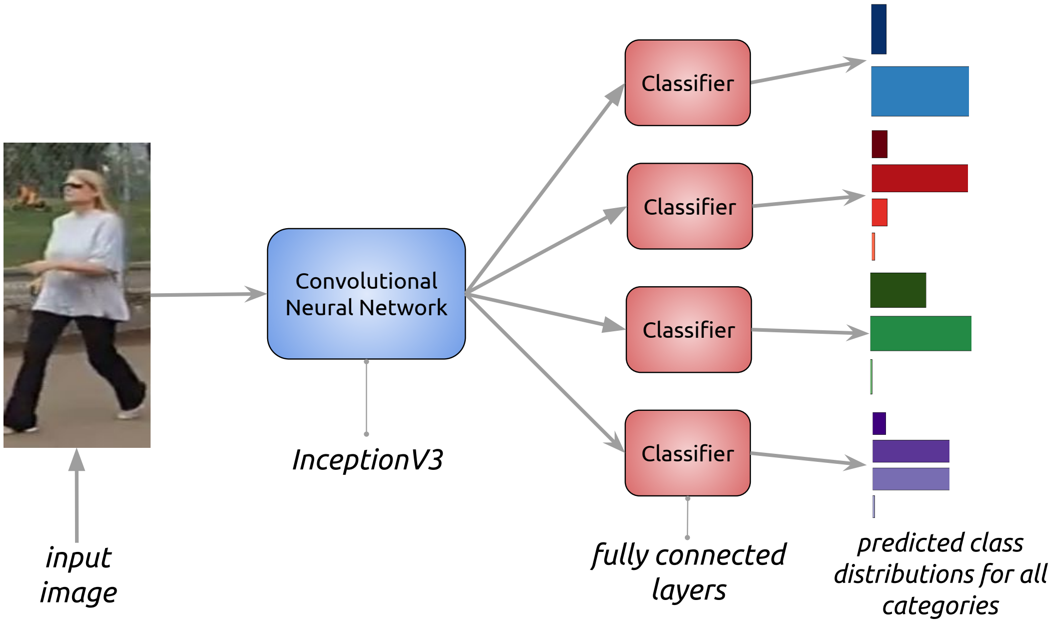


Figure 4.12: Unified Model. The *unified* model takes an input image which is passed through a convolutional neural network with an InceptionV3 [51] architecture, pre-trained on ImageNet [26]. The final 1000-way fully-connected classification layer of the InceptionV3 model is then replaced with four separate fully-connected classification layers — one for each of the fine-grained categories in the dataset.

model benchmarked in the published version of this work.

In this framework, features are extracted by applying deep convolutional nets to image regions that are normalised by pose. To elaborate, given a sample, image regions are extracted and then warped so that they are aligned to a set of prototypical models. Each warped region is then fed through a deep convolutional network where features are extracted from certain layers. The features from each warped region are then concatenated and used in a classifier.

For this benchmark, given an image containing a person, two regions are extracted: 1) the full bounding box and 2) the full bounding box warped to a hand-defined pose prototype using a similarity alignment model, which was suggested by Branson *et al*. to work best [7]. The pose prototype only considers the location of the shoulders, the hips and the head. The ground-truth keypoints for the left and right shoulder, the left and right hip, the top of the head, and the chin are used to compute the warping of the input image to the prototype.

The two regions are then fed through a convolutional neural network with an

AlexNet [30] architecture that has been pre-trained on ImageNet [26]. Features are extracted from the 5th layer after max-pooling (this layer gave the best performance). Features for each region are then concatenated and separate one-vs-all linear SVMs are trained to classify each of the fine-grained categories. To be clear, the only trainable weights are those of the SVMs; the weights of the neural network are fixed. This method is referred to as “pose: bb+body”. We also consider the case where only the bounding box is used as the extracted image region (the warped input image is not used). This method is referred to as “pose: bb”. A diagram of this baseline model can be found in Figure 4.11.

Unified Model

Since the original version of this work was published, neural network architectures have improved. We thus propose a *unified* model to predict fine-grained categories of people. The unified model takes an input image which is passed through a convolutional neural network with an InceptionV3 [51] architecture that has been pre-trained on ImageNet [26].

The final 1000-way fully-connected classification layer of the InceptionV3 model is then replaced with four separate fully-connected classification layers — one for each of the fine-grained categories we are interested in. The number of neurons in each of the classification layers is equivalent to the number of classes for a particular fine-grained category. The resulting model thus takes an image and outputs the class distributions for each of the fine-grained categories of interest.

During training the network is fine-tuned so the weights of the convolutional stage as well as the classifier stages can be updated. The categorical cross-entropy loss is used for each of the four outputs. The total loss is the sum of each of the individual losses (this could be a weighted sum, giving different importance to different categories, but it is something we have not explored). This model differs from the baseline model in that there is no need for pose and the model can be trained end-to-end. A diagram of the unified model can be found in Figure 4.12.

Results: Single Images of People

The class average classification accuracy (the accuracy for each class is computed independently then averaged across all the classes for a given category) is reported for each of the fine-grained categories in the dataset. This metric says that predicting one class correctly is equally as important as predicting any of the other classes correctly. For all the experiments, keypoint occlusion information was not used and

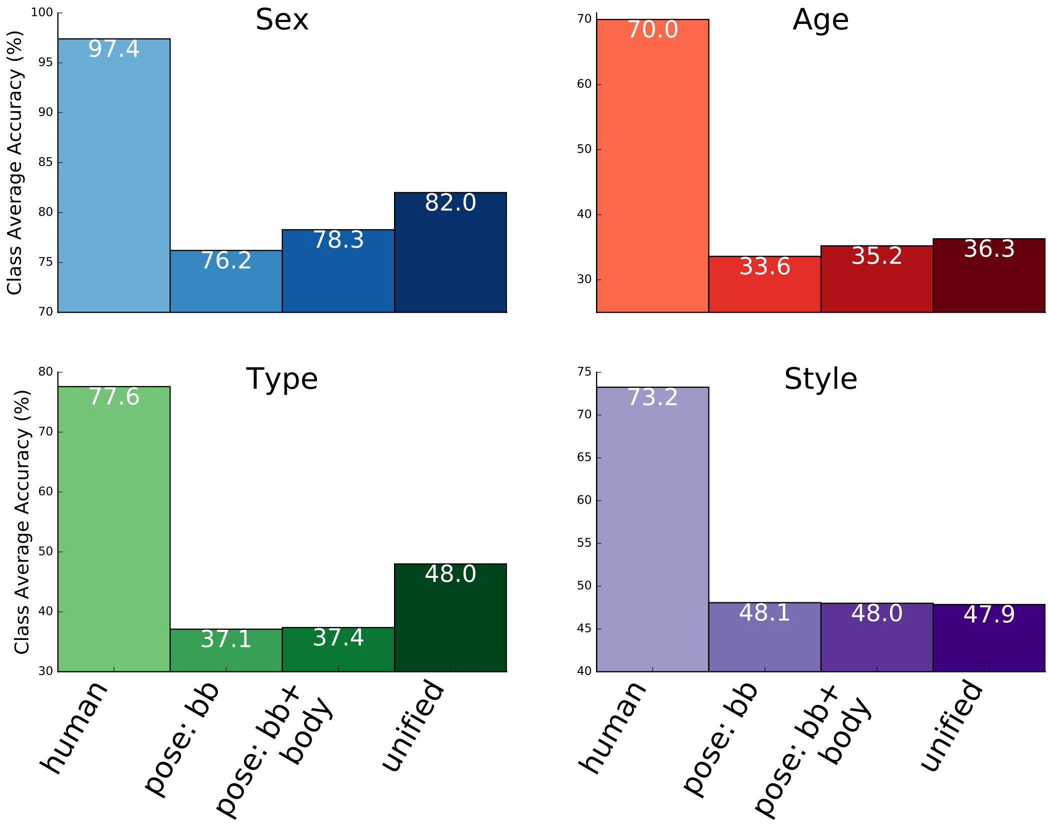


Figure 4.13: Single Image Fine-grained Classification Results. The class average accuracy is reported for each of the subcategories comparing each of the three models to human performance. Compared to humans, classification performance is quite poor for age, body type and style. Sex classification is better. This suggests that this is a challenging dataset to work with. The unified model does only marginally better or about the same when compared to the baseline models for sex, age and style. For body type the unified model does 10% better than the pose normalised models.

samples with missing parts were ignored.

The baseline model was trained using code that was obtained directly from the authors [7]. The unified model is implemented using *keras* [12] with *tensorflow* [1] as a backend. We use stochastic gradient descent for training with a learning rate of 0.0001, a momentum of 0.9 and a batch size of 32. Weight decay is used for regularisation with a value of 0.0005. Dropout is also used between the convolutional layers and the classification layers with a drop ratio of 0.5. A validation set is used to determine when to stop training (the best validation loss was achieved after 28 epochs). Finally, since the classes for each of the fine-grained categories are imbalanced, the dataset is *balanced* during training by giving each

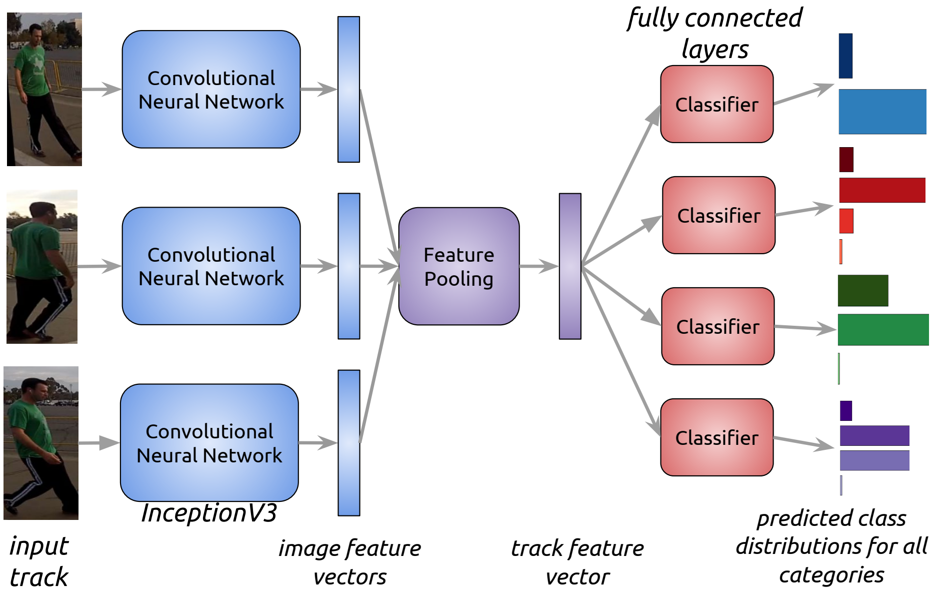


Figure 4.14: Fine-grained Classification Model for Sequences of Images. Features are extracted independently for each image using a convolutional network. The image feature vectors are combined or pooled to form a track feature vector. The prediction is then made using the track feature vector. Image features can be pooled in a number of different ways. Refer to Section 4.7 for details.

class a weight that is inversely proportional to the frequency of that class.

The input images are also augmented during training with a random horizontal flip, a random translation of up to 10% of the image's height and width, a random rotation of up to 10°, a random shear of up to 5°, a random zoom between 0.9 and 1.1, and a random channel shift between -10 and 10 for each channel.

For reference we also compare to human performance.

Results can be found in Figure 4.13. Compared to humans, classification performance is poor for age, body type and style, with the best performing model being 35%, 30% and 25% worse respectively. Sex classification is only 15% poorer than humans. The unified model does only marginally better or about the same when compared to the baseline models for sex, age and style. The greatest difference in performance is for body type with the unified model doing 10% better than the pose normalised models. This suggests that this is a challenging dataset.

4.7 Experiments: Fine-Grained Classification - Sequences of Images

It can be challenging to determine the sex, age, body type or style of a person from a single cropped out window of that person. The task is easier if a sequence of images of that person is given from different viewpoints (front-view, side-view, rear-view). This is the reason humans annotators were given a sequence of four images when labelling fine-grained categories (see Figure 4.24 in the supplementary materials to see an example of a sequence human annotators were given).

In the previous section, comparing the machines performance on single images to humans may be unfair since humans had the advantage of seeing a sequence of images to make their prediction. In this section we instead give the machine a sequence of images as input. We also explore different strategies for combining information from a sequence of images of the same person to obtain a consistent fine-grained classification for that person.

Combining Information from a Sequence of Images

In this section we assume that a unified model has already been trained on single images. This model will be referred to as the *single-image model*.

There are a number of ways to combine information from a sequence of images of a particular person. The simplest way to obtain a consistent fine-grained classification for that individual is to predict the class for each image independently using the *single-image model* then take a majority vote. We will refer to this method as **vote**.

An alternative approach as shown in Figure 4.14 is to independently extract features for each image using the convolutional network of the *single-image model* (these weights are fixed for all variants). Combine or pool the image feature vectors into a new, *track* feature vector. Then classify the track feature vector using the classifiers of the *single-image model*.

There are a number of alternatives for pooling the image features:

1. Average Pooling. The track feature vector is the mean of the image feature vectors. If the classifier weights of the *single-image model* are fixed we refer to this variant as **ave**. If they can be fine-tuned this variant is referred to as **ave+ft**. This is similar to the late pooling used by Ng *et al.* [41] for activity recognition.

2. Max Pooling. The track feature vector is the component-wise maximum of the image feature vectors. The variants are known as **max** and **max+ft** as explained above.

3. Fully Connected. The track feature vector is a weighted sum of the image feature vectors followed by a non-linearity. The weights are learnt. The classifier weights of the *single-image model* are tuneable. This variant is referred to as **fc**. This is similar to the late fusion used by Karpathy *et al*. [41] for activity recognition.

4. RNN. The image feature vectors are sequentially combined using an RNN. The track feature vector is the output of an RNN at its final time step. We consider RNNs with 5 (**rnn 5**) and 10 (**rnn 10**) time steps. Again the classifier weights of the *single-image model* are tuneable. This is similar to the LCRN method proposed by Donahue *et al*. [17] for activity recognition.

Results: Tracks of People

For the **rnn** experiments Gated Rectified Units [11] (GRUs) with 32 units are used as the recurrent neural network. For the **fc**, **ave+ft** and **max+ft** experiments the input sequence length was a maximum of five images. If an image sequence was longer than five images it was linearly sampled in time to obtain five images.

Experiments were run using each of the previously mentioned methods for pooling for each of the subcategories in the dataset, as was done in the single image case. Results can be found in Figure 4.15. For conciseness we report only the class average accuracy in the main document. Per-class results can be found in Figures 4.26, 4.27, 4.28 and 4.29 in the supplementary material. We compare to single image performance using the joint model.

For sex, all sequence based methods improve class average accuracy when compared to the single-image method. Average pooling with no fine tuning of the classification stage sees the greatest improvement (6.7%). For age, **max** and **fc** do worse than the single-image method while the rest do marginally better. The best being **rnn 10** with an improvement of 2.6%. For body type, all methods do significantly worse than the single-image method. For style, **vote** is the best performer with a .9% improvement.

From the class average accuracy results it is difficult to draw any conclusions about how information from sequences of images should be combined. If instead we look at the overall accuracy results (simply count the number of correct predictions regardless of class) as shown in Figure 4.16, a far more consistent result emerges. Now, all sequence methods do better than the single image method for each of the fine-grained categories. The best method is **max** with a 8.4%, 10.6% and 6.3%

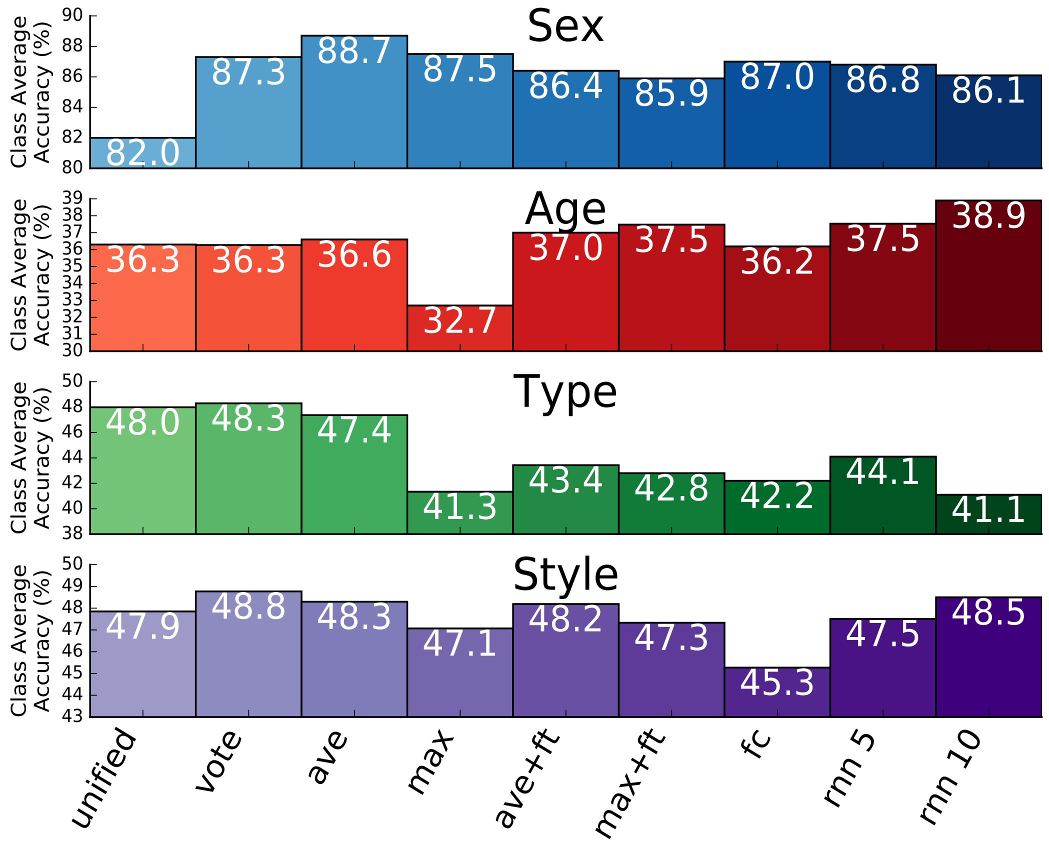


Figure 4.15: Fine-grained class average accuracy results using sequences of images. Experiments predicting the sex, age, body type and style of a person were performed using each of the feature pooling variants mentioned in Section 4.7. A comparison is made to single image performance using the **joint** model. For sex, all sequence based methods improve class average accuracy when compared to the single-image method. Average pooling with no fine tuning of the classification stage sees the greatest improvement (6.7%). For age and style most sequence-based methods do better than the single-image method while for body type they always does worse.

improvement for age, type and style respectively. For sex there is a 5.7% increase using **max** but a 6.6% using **ave**.

To make further conclusions the per-class results in Figures 4.26-4.29 need to be looked at. The increase in overall accuracy and the fluctuations in class average accuracy for the age, style and body type categories indicate that the sequence-based models are over-fitting to the most common class since these categories are the most imbalanced. This behaviour is best highlighted by the **max** and **fc** methods which overfit the most to the most common classes — healthy, athletic and middle aged accuracy all increase while the other less represented classes all decrease in

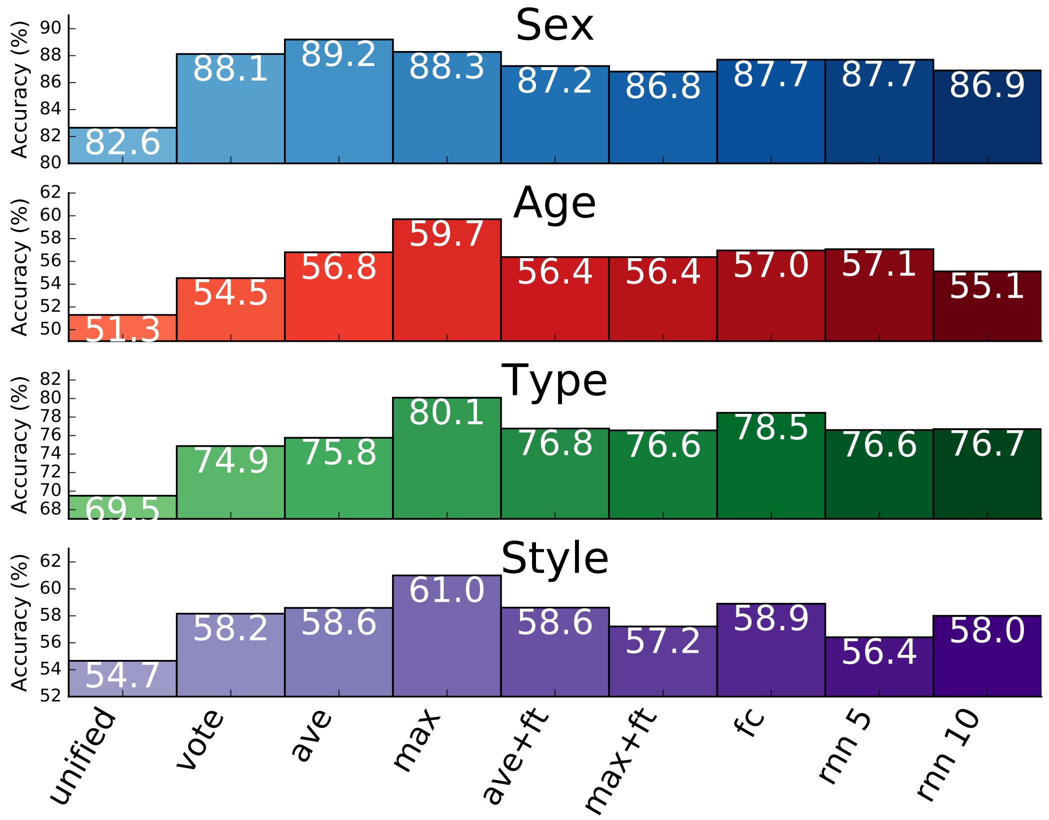


Figure 4.16: Fine-grained overall accuracy results using sequences of images. The overall accuracy counts the number of correct predictions regardless of class. All sequence methods do better than the single image method for each of the fine-grained categories. However, since the age, type and style classes are all highly unbalanced, these results, when looked at in conjunction with the results of Figure 4.15, suggest that the models are probably over-fitting to the most common class of a category. Per-class accuracy results can be found in 4.11 to support this claim.

accuracy.

Another observation is that the severely under-represented classes (dressy for style, child and teen for age, and under for body type (see Figure 4.7 for the class distributions for each fine-grained category)) are effected the most by using sequences of images. Most methods cause the accuracy to drop to zero or close to zero. Since images are being combined into sequences, the number of training examples for the under-represented classes is further reduced and so they essentially get ignored during training (even when training examples are weighted to account for unbalanced classes). Humans still perform extremely well on the under-represented classes.

Figure 4.17 shows results on age, body type and style for the case where the severely

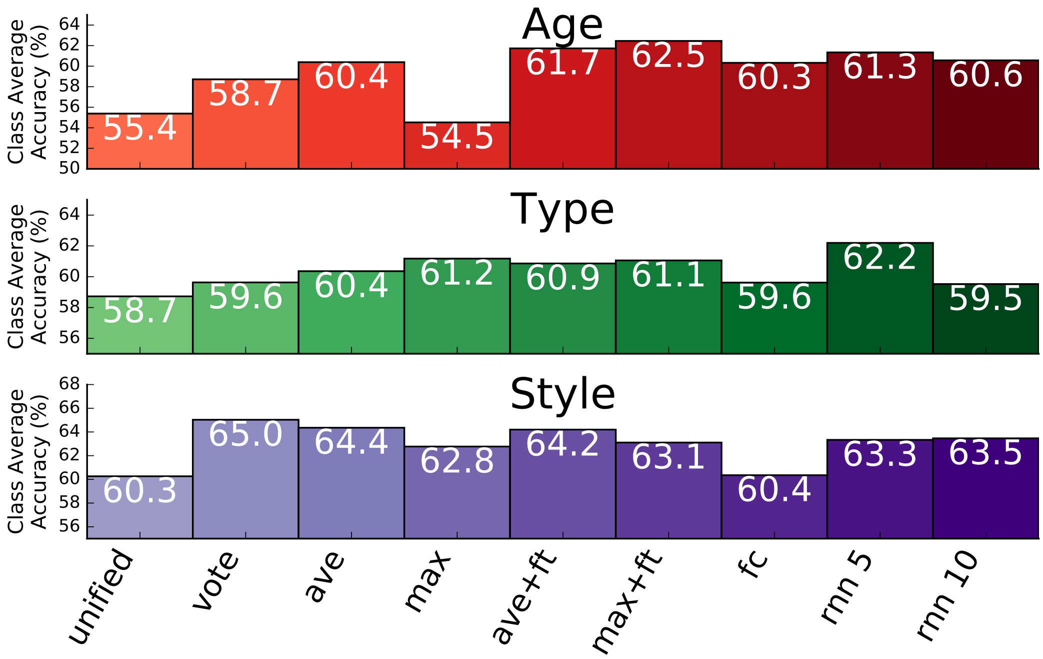


Figure 4.17: Fine-grained class average accuracy results using sequences of images, re-computed to ignore severely under-represented classes. The ignored classes correspond to dressy for style, child and teen for age, and under for body type. The sequence-based methods nearly always have a better class average accuracy than the single-image method, **joint**. The best pooling method still varies between categories.

under-represented classes are ignored and the class average accuracy is re-computed. A more consistent result is now seen with the sequence-based methods nearly always having a better class average accuracy than the single-image method. The best pooling method still varies between categories with the best results for age, body type and style coming from **max+ft**, **rnn 5** and **vote** respectively.

To summarise, for the purpose of fine-grained classification, combining information from a sequence of images of an individual then predicting the label[3] is 3.5-7.1% better, in terms of class average accuracy, than independently predicting the label of each image, assuming that severely under-represented classes are ignored (a severely under-represented class makes up less than 5% of the class labels). The best method for combining information is unclear from these experiments with all variants performing differently depending on the fine-grained category.

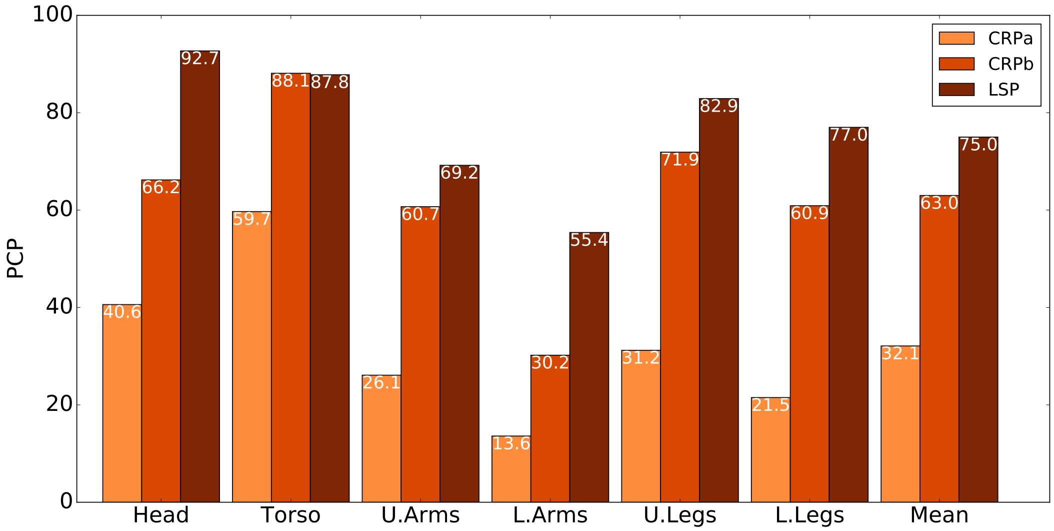


Figure 4.18: Pose estimation results. We report the PCP for the parts in our dataset using the method described in Section 4.8. CRPa corresponds to the pose model trained using the LSP dataset [27] but tested on CRP. CRPb correponds to the case when the CRP dataset is used for both training and testing. Performance is best on the torso, with the lower arms and lower legs performing the most poorly. For comparison, results for a pose model trained and tested using the LSP dataset — the current pose estimation benchmark — are provided. The errors for our dataset are worse than those on the LSP dataset. This suggests that ours is a challenging dataset for pose estimation.

4.8 Experiments: Pose Estimation

Since many fine-grained classification techniques rely on parts, it is important to look at pose estimation. To benchmark human pose estimation we used the state-of-the-art, articulated pose estimator of Chen and Yuille [10]. This method extends Yang and Ramanan's work [55] to use deep features. The code is publicly available.

Two experiments were run using a single train/test split. In the first experiment, the pose model was trained using data from the LSP dataset [27]. In the second, training was done using data from the CRP dataset. In both cases keypoint occlusion information was not used and samples with missing parts were ignored.

The results are reported using a standard measure: the stricter interpretation of the percentage of correct parts (PCP) [46]. The results are shown in Figure 4.18

The results indicate that this dataset is more challenging than the existing pose estimation benchmark, the LSP dataset. The mean PCP is approximately 12% lower

[3] the predicted label is then assigned to each of the images in the sequence.

than that of LSP. The high amount of occlusion present in our dataset is contributing to the poorer results.

4.9 Discussion and Conclusions

We introduce a video dataset designed to study fine-grained classification of people using the entire human body. Its novel and distinctive features are size, realism (natural behaviour, variety in viewpoint, moving camera), fine-grained multi-label attributes (sex, weight, clothing, age), detailed annotations, and public availability. The dataset is called Caltech Roadside Pedestrians (CRP).

Three sets of experiments were conducted to provide a performance baseline for the dataset. The first was a fine-grained classification task using single images, where we introduced a unified model based on the InceptionV3 [51] neural network architecture. This model takes a single image as input and outputs the class distributions for each of the four fine-grained categories in the CRP dataset. The unified model is compared to the pose normalisation + deep network system of Branson *et al.* [7], which has state-of-the-art performance on bird species classification. The unified model does only marginally better or about the same when compared to the baseline model for sex, age and style. For body type the unified model does 10% better than the pose normalised model. Compared to humans, classification performance is quite poor for age, body type and style (29.4-33.7% worse). Sex classification performance is better (only 15.4% worse). This suggests that our realistic and large dataset is challenging and will contribute to advance the state-of-the-art in fine-grained classification.

The second set of experiments was a fine-grained classification task using sequences of images where we explore different strategies for combining information so that a consistent fine-grained classification prediction is obtained. The results indicate that combining information from a sequence of images of an individual and then predicting the label is 3.5-7.1% better, in terms of class average accuracy, than independently predicting the label of each image, assuming that severely under-represented classes are ignored. A class is considered to be severely under-represented if it makes up less than 5% of the class labels. The best method for combining information is unclear from these experiments with all variants performing differently depending on the fine-grained category.

The third set of experiments was a pose estimation task where we used an articulated pose model with deep features as a baseline method [10]. We find that the baseline method performance is 12% lower than the same methods performance on LSP [27], an existing pose estimation dataset. This suggests that CRP will also contribute to advance the state-of-the-art in pose estimation.

An interesting feature of the dataset is the presence of severely under-represented classes for some of the fine-grained categories. While human performance at predicting the class labels of images from these classes is still quite high, achieving the same level of accuracy from a small number of examples is still a challenge for computer vision algorithms. While the comparison may not be completely fair since humans have the advantage of seeing many more training examples throughout their lifetime, the ability of humans to apply what they know about one class to another class is unmatched by machines. Exploring better methods for transfer learning may be the first step towards improving the fine-grained classification performance on the under-represented classes.

Acknowledgements

This work is funded by the ARO-JPL NASA Stennis NAS7.03001 grant and the Gordon and Betty Moore Foundation.

4.10 Appendix: Dataset Collection

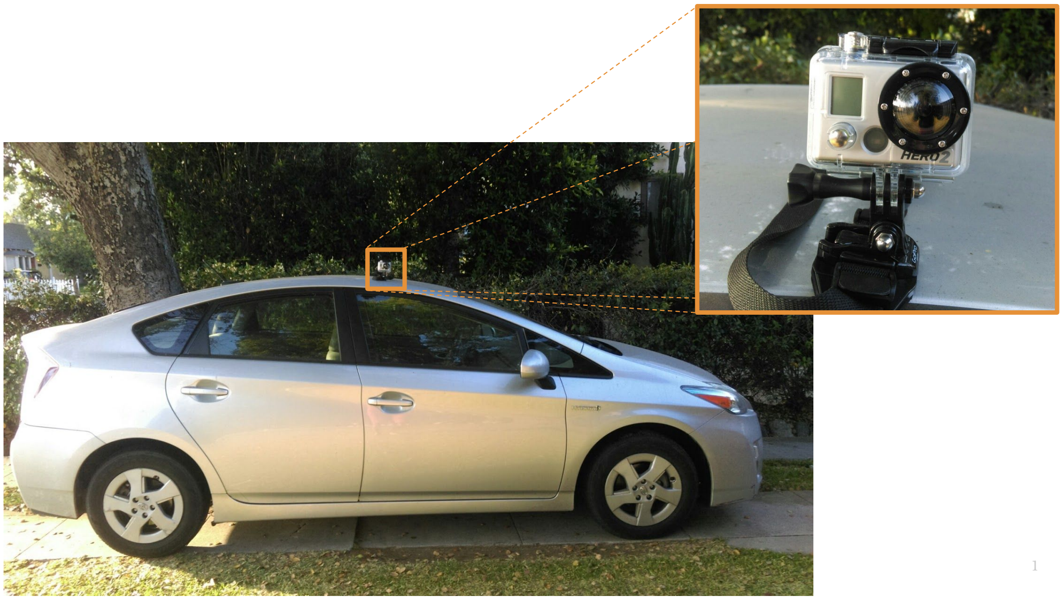


Figure 4.19: The GoPro camera attached to the vehicle.

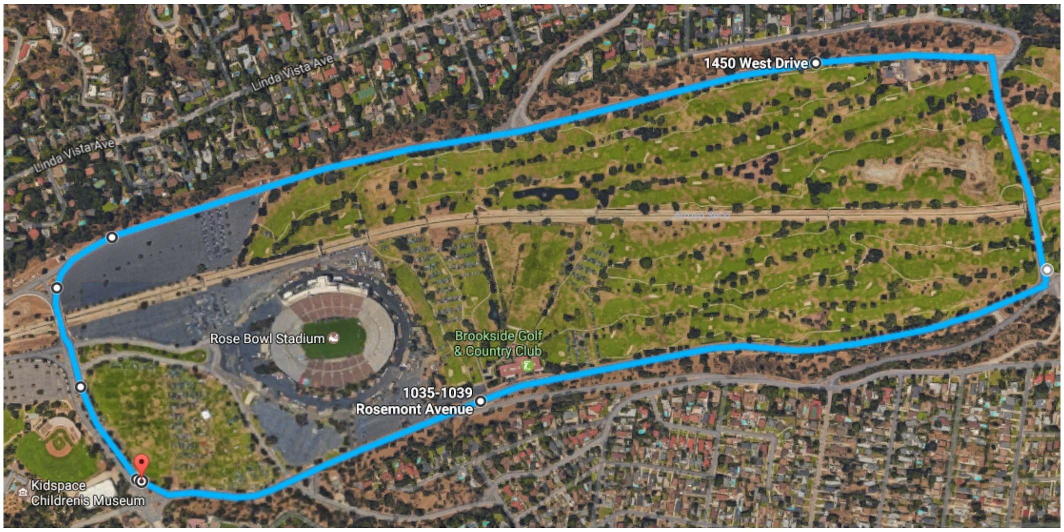


Figure 4.20: The route that was taken during capture. The circuit was completed three times in a session. A total of seven sessions were captured on different days.

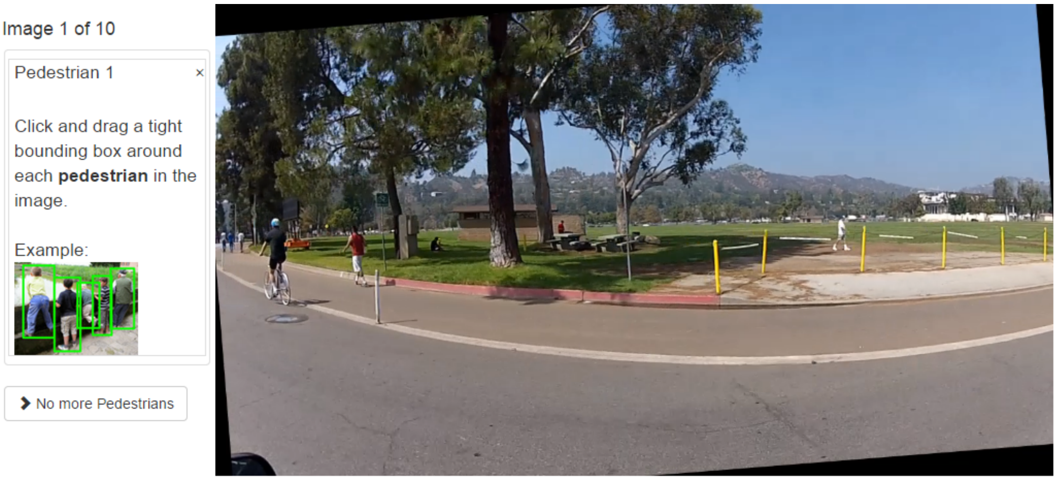


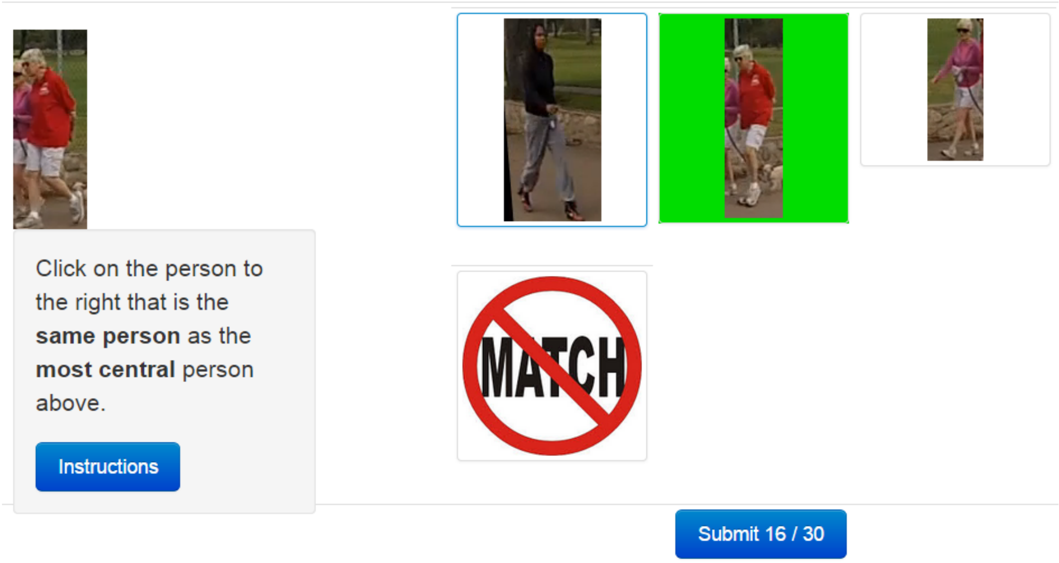
Image 1 of 10

Pedestrian 1

Click and drag a tight bounding box around each **pedestrian** in the image.

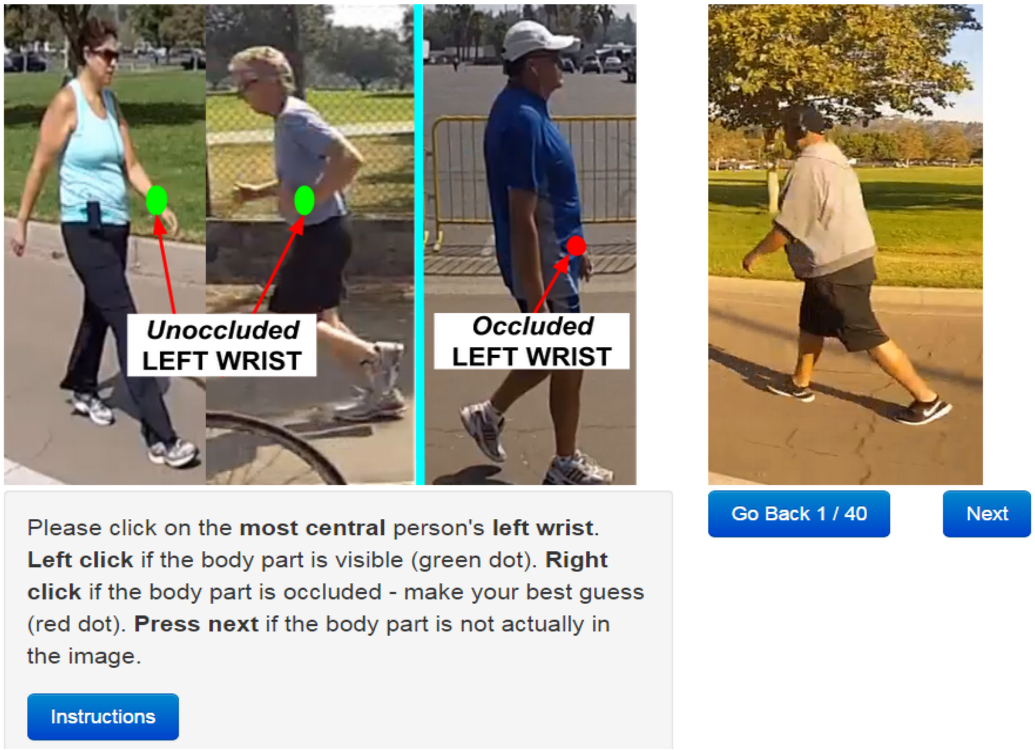
Example:

Figure 4.21: The interface used by Amazon Mechanical Turk workers to draw bounding boxes around people in the Caltech Roadside Pedestrians Dataset. This interface was created by Dr. Steve Branson.



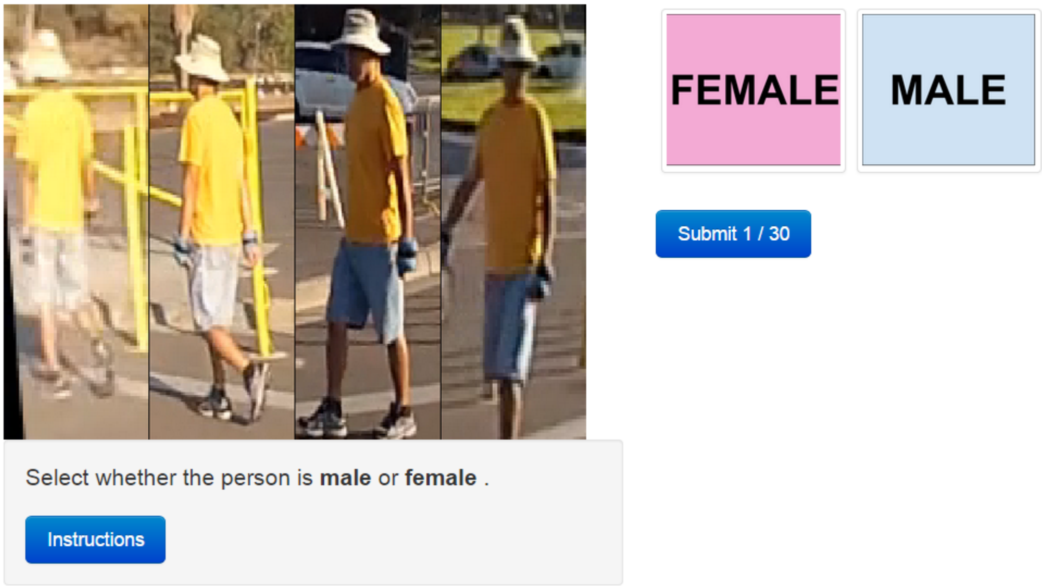
Click on the person to the right that is the **same person** as the **most central** person above.

Figure 4.22: The interface used by Amazon Mechanical Turk workers to create tracks of people in the Caltech Roadside Pedestrians Dataset.



Please click on the **most central** person's **left wrist**. **Left click** if the body part is visible (green dot). **Right click** if the body part is occluded - make your best guess (red dot). **Press next** if the body part is not actually in the image.

Figure 4.23: The interface used by Amazon Mechanical Turk workers to label the left elbow of a person in the Caltech Roadside Pedestrians Dataset. The other body parts are collected in a similar way.



Select whether the person is **male** or **female** .

Figure 4.24: The interface used by Amazon Mechanical Turk workers to label the sex attribute of a person in the Caltech Roadside Pedestrians Dataset. The other attributes are collected using a similar interface.

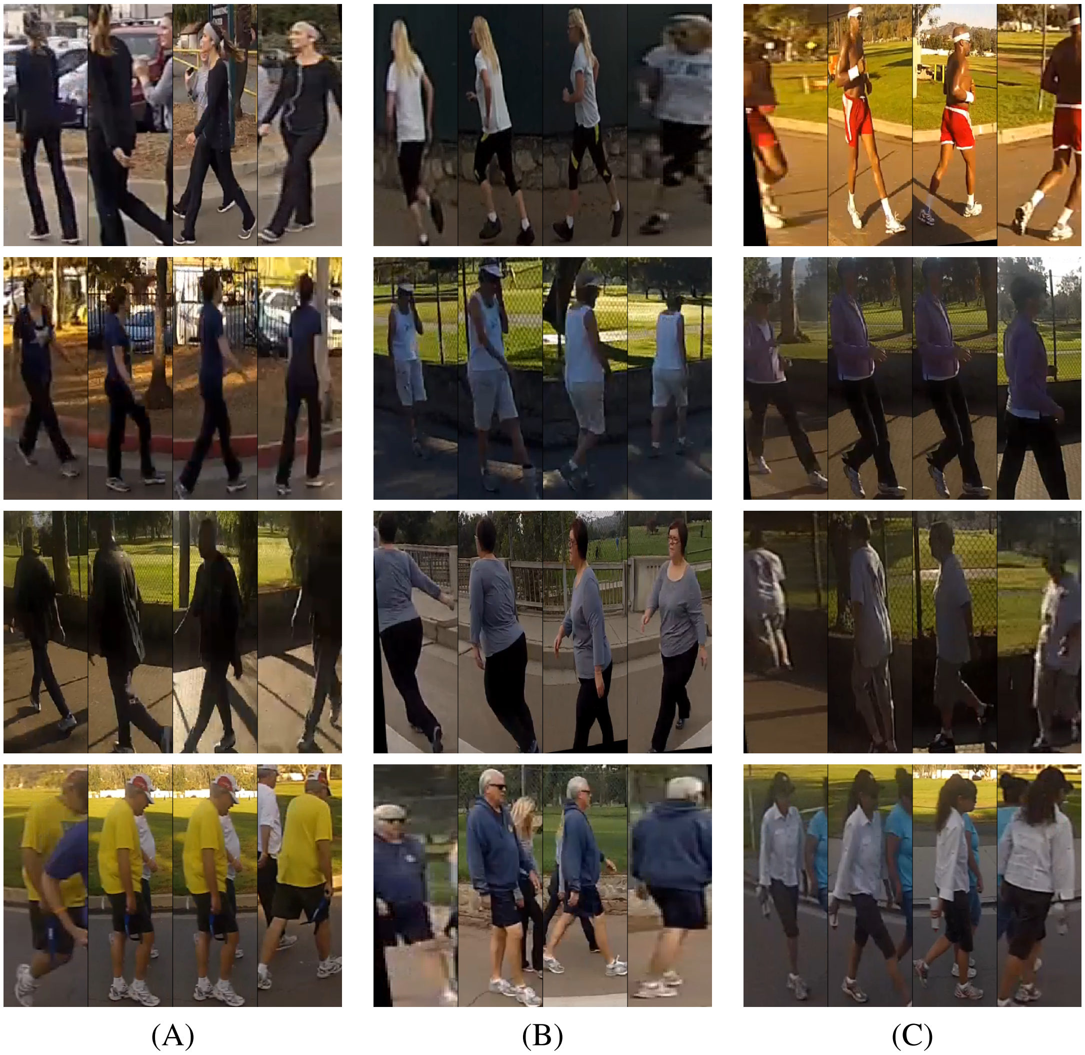


Figure 4.25: Class examples. From top to bottom the corresponding class labels for each category are: (A) age - teen, young adult, middle aged, and senior; (B) weight - under, healthy, over, and over; (C) clothing style - workout, light athletic, casual comfort and dressy. The examples shown in this figure illustrate the variety of **l**ighting conditions (full sunlight, hazy, front and back lighting), **v**iewpoints (front, profile, back) and **b**ackgrounds.

4.11 Appendix: Fine Grained Classification Results

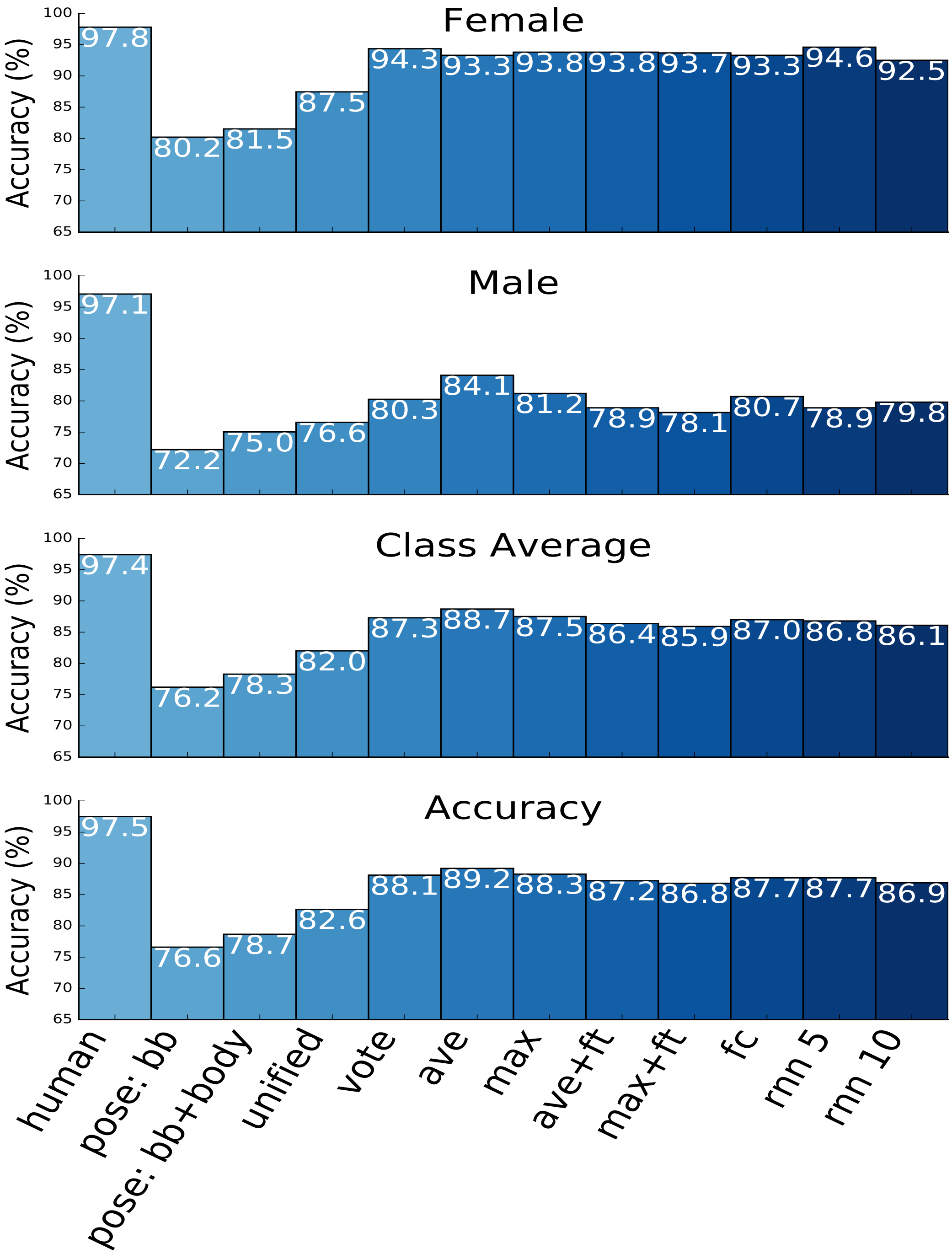


Figure 4.26: Per-class fine-grained classification results for the Sex category.

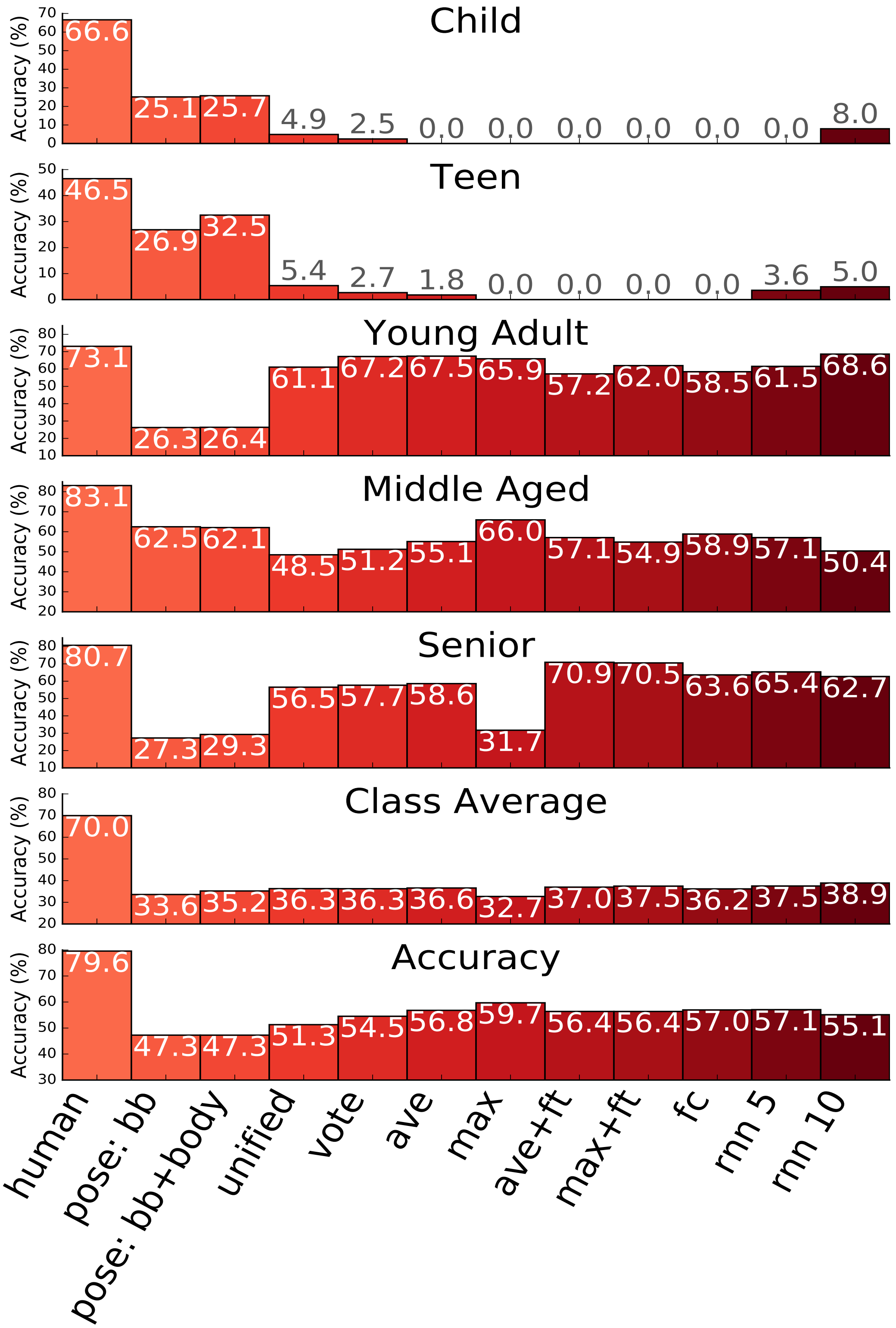


Figure 4.27: Per-class fine-grained classification results for the Age category.

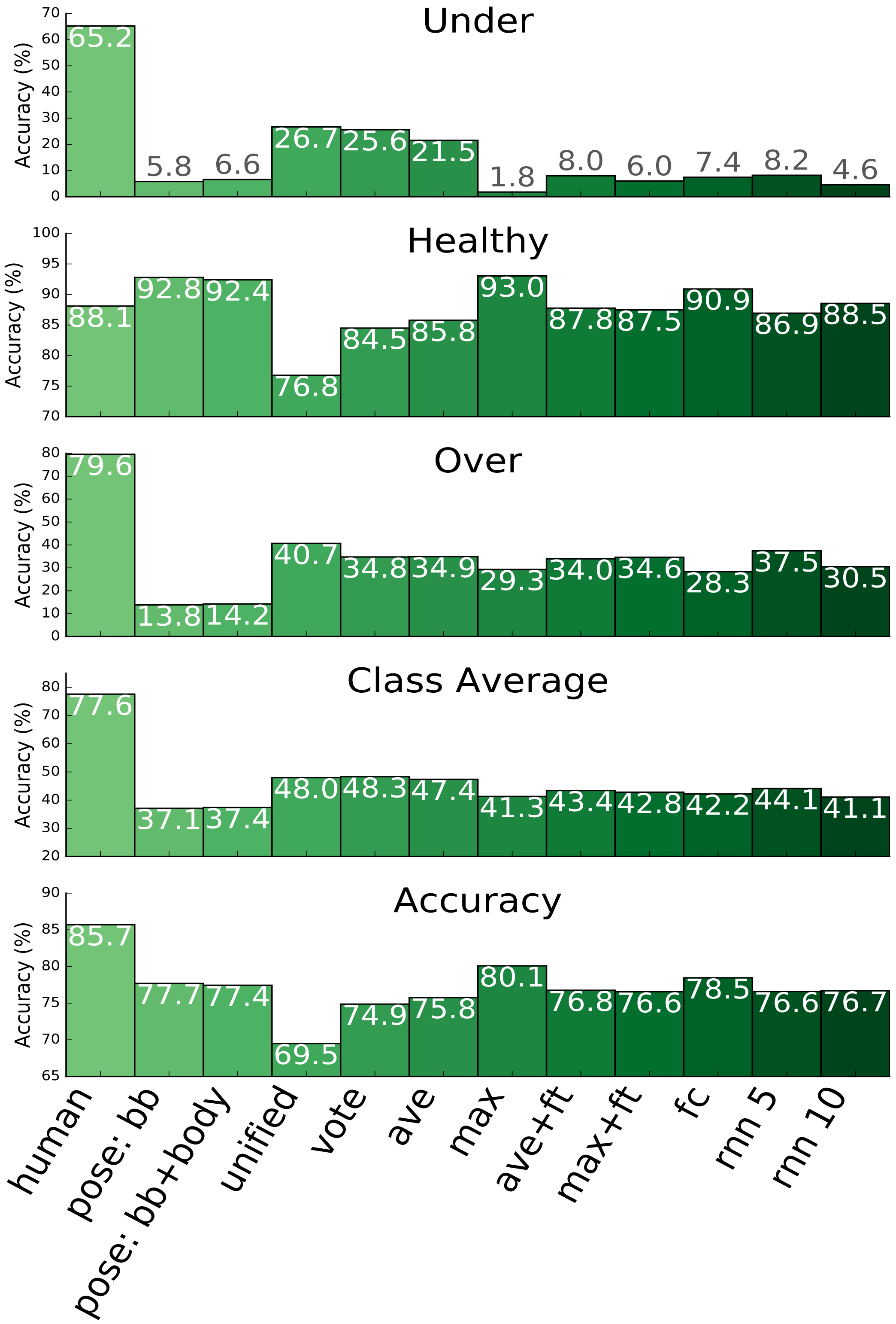


Figure 4.28: Per-class fine-grained classification results for the Body Type category.

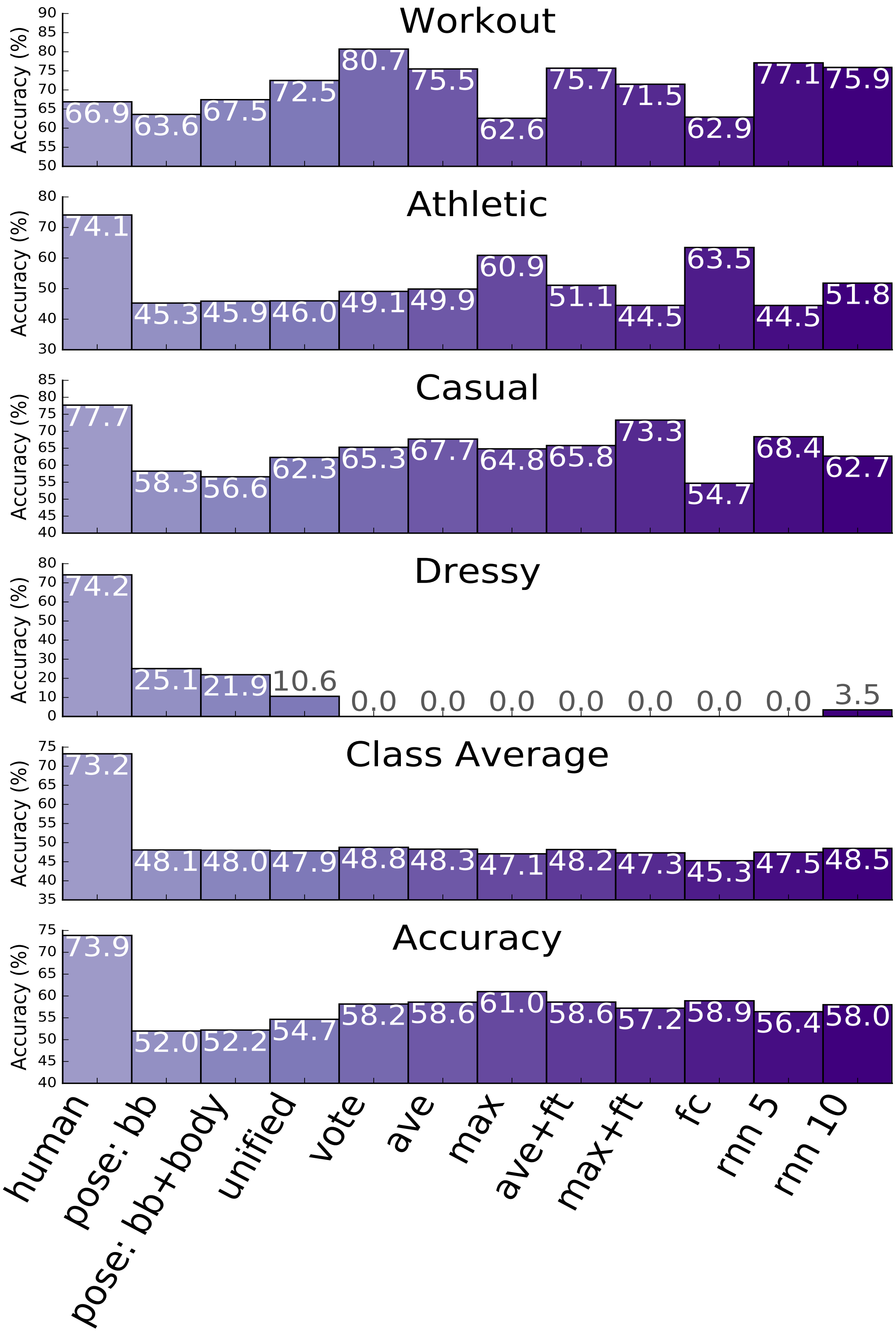


Figure 4.29: Per-class fine-grained classification results for the Style category.

References

[1] M. Abadi. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. 2015.

[2] M. Baccouche et al. “Sequential Deep Learning for Human Action Recognition”. In: *Human Behavior Understanding*. 2011, pp. 29–39. DOI: 10.1007/978-3-642-25446-8_4.

[3] S. Baluja and H. Rowley. “Boosting Sex Identification Performance”. In: *International Journal of Computer Vision* 71.1 (June 2006), pp. 111–119. DOI: 10.1007/s11263-006-8910-9.

[4] K. Bashir, T. Xiang, and S. Gong. “Gait Recognition Using Gait Entropy Image”. In: *ICDP*. 2009. DOI: 10.1049/ic.2009.0230.

[5] T. Berg et al. “Birdsnap: Large-Scale Fine-Grained Visual Categorization of Birds”. In: *CVPR*. 2014. DOI: 10.1109/CVPR.2014.259.

[6] L. Bourdev, S. Maji, and J. Malik. “Describing People: A Poselet-Based Approach to Attribute Classification”. In: *ICCV*. 2011. DOI: 10.1109/ICCV.2011.6126413.

[7] S. Branson et al. “Improved Bird Species Recognition Using Pose Normalized Deep Convolutional Nets”. In: *BMVC*. 2014. DOI: 10.5244/C.28.87.

[8] S. Branson et al. “Visual Recognition with Humans in the Loop”. In: *ECCV*. 2010. DOI: 10.1007/978-3-642-15561-1_32.

[9] L. Cao et al. “Gender Recognition from Body”. In: *ACM*. 2008. DOI: 10.1145/1459359.1459470.

[10] X. Chen and A. Yuille. “Articulated Pose Estimation by a Graphical Model with Image Dependent Pairwise Relations”. In: *NIPS*. 2014. DOI: 10.1.1.672.1960.

[11] K. Cho et al. “Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation”. In: *EMNLP*. 2014.

[12] F. Chollet et al. *Keras*. https://github.com/fchollet/keras. 2015.

[13] M. Collins, J. Zhang, and P. Miller. “Full Body Image Feature Representations for Gender Profiling”. In: *ICCV Workshop*. Sept. 2009. DOI: 10.1109/ICCVW.2009.5457467.

[14] G. Cottrell and J. Metcalfe. “EMPATH: Face, Emotion, and Gender Recognition using Holons”. In: *NIPS*. 1990.

[15] J. Deng et al. “ImageNet: A Large-Scale Hierarchical Image Database”. In: *CVPR*. 2009. DOI: 10.1109/CVPR.2009.5206848.

[16] P. Dollar et al. “Pedestrian Detection: A Benchmark”. In: *CVPR*. 2009. DOI: 10.1109/CVPR.2009.5206631.

[17] J. Donahue et al. “Long-term Recurrent Convolutional Networks for Visual Recognition and Description”. In: *CVPR*. 2015. DOI: 10.1109/TPAMI.2016.2599174.

[18] S. Fu, H. He, and Z.-G. Hou. “Learning Race from Face: A Survey”. In: *PAMI* 36.12 (2014), pp. 2483–2509. DOI: 10.1109/TPAMI.2014.2321570.

[19] Y. Fu, G. Guo, and T. S. Huang. “Age Synthesis and Estimation via Faces: A Survey.” In: *PAMI* 32.11 (Nov. 2010), pp. 1955–76. DOI: 10.1109/TPAMI.2010.36.

[20] B. Golomb, D. Lawrence, and T. Sejnowski. “SEXNET: A Neural Network Identifies Sex From Human Faces.” In: *NIPS*. 1990.

[21] D. Gray and H. Tao. “Viewpoint Invariant Pedestrian Recognition with an Ensemble of Localized Features”. In: *ECCV*. 2008. DOI: 10.1007/978-3-540-88682-2_21.

[22] J. Han and B. Bhanu. “Individual Recognition Using Gait Energy Image.” In: *PAMI*. Vol. 28. 2. Mar. 2006, pp. 316–22. DOI: 10.1109/TPAMI.2006.38.

[23] S. Herath, M. Harandi, and F. Porikli. “Going Deeper into Action Recognition: A Survey”. In: *Image and Vision Computing* 60 (2017), pp. 4–21. DOI: 10.1016/j.imavis.2017.01.010.

[24] H. Iwama et al. “The OU-ISIR Gait Database Comprising the Large Population Dataset and Performance Evaluation of Gait Recognition”. In: *IEEE Trans. Inf. Forensics Security* 7.5 (Oct. 2012), pp. 1511–1521. DOI: 10.1109/TIFS.2012.2204253.

[25] S. Ji et al. “3D Convolutional Neural Networks for Human Action Recognition”. In: *PAMI* 35.1 (Jan. 2013), pp. 221–231. DOI: 10.1109/TPAMI.2012.59.

[26] Y. Jia et al. “Caffe : Convolutional Architecture for Fast Feature Embedding”. In: *ACM Conference on Multimedia*. 2014. DOI: 10.1145/2647868.2654889.

[27] S. Johnson and M. Everingham. “Clustered Pose and Nonlinear Appearance Models for Human Pose Estimation”. In: *BMVC*. 2010. DOI: 10.5244/C.24.12.

[28] A. Karpathy et al. “Large-Scale Video Classification with Convolutional Neural Networks”. In: *CVPR*. 2014. DOI: 10.1109/CVPR.2014.223.

[29] A. Khosla et al. “Novel Dataset for Fine-Grained Image Categorization: Stanford Dogs”. In: *FGVC Workshop, CVPR*. 2011.

[30] A. Krizhevsky, I. Sutskever, and G. E. Hinton. “ImageNet Classification with Deep Convolutional Neural Networks”. In: *NIPS*. 2012.

[31] N. Kumar, P. Belhumeur, and S. Nayar. “FaceTracer: A Search Engine for Large Collections of Images with Faces”. In: *ECCV*. 2008. DOI: 10.1007/978-3-540-88693-8_25.

[32] N. Kumar et al. “Leafsnap: A Computer Vision System for Automatic Plant Species Identification”. In: *ECCV*. 2012. DOI: 10.1007/978-3-642-33709-3_36.

[33] T.-Y. Lin et al. “Microsoft COCO: Common Objects in Context”. In: *ECCV*. 2014. DOI: 0.1007/978-3-319-10602-1_48.

[34] J. Liu et al. “Dog Breed Classification Using Part Localization”. In: *ECCV*. 2012. DOI: 10.1007/978-3-642-33718-5_13.

[35] K. Liu et al. “A Spatio-Temporal Appearance Representation for Video-Based Pedestrian Re-Identification”. In: *ICCV*. 2015. DOI: 10.1109/ICCV.2015.434.

[36] S. Maji et al. *Fine-Grained Visual Classification of Aircraft*. Tech. rep. 2013.

[37] Y. Makihara et al. “The OU-ISIR Gait Database Comprising the Treadmill Dataset”. In: *IPSJ Transactions on Computer Vision and Applications* 4 (2012), pp. 53–62. DOI: 10.2197/ipsjtcva.4.53.

[38] G. Mart et al. “Dictionary-Free Categorization of Very Similar Objects via Stacked Evidence Trees”. In: *CVPR*. 2009. DOI: 10.1109/CVPR.2009.5206574.

[39] N. McLaughlin, J. M. d. Rincon, and P. Miller. “Recurrent Convolutional Network for Video-Based Person Re-identification”. In: *CVPR*. 2016. DOI: 10.1109/CVPR.2016.148.

[40] B. Moghaddam and M.-H. Yang. “Learning Gender With Support Faces”. In: *PAMI* 24.5 (May 2002), pp. 707–711. DOI: 10.1109/34.1000244.

[41] J. Y.-H. Ng et al. “Beyond Short Snippets: Deep Networks for Video Classification”. In: *CVPR*. 2015. DOI: 10.1109/CVPR.2015.7299101.

[42] M.-E. Nilsback and A. Zisserman. “Automated Flower Classification over a Large Number of Classes”. In: *ICCVGIP*. 2008. DOI: 10.1109/ICVGIP.2008.47.

[43] M.-E. Nilsback and A. Zisserman. “A Visual Vocabulary for Flower Classification”. In: *CVPR*. 2006. DOI: 10.1109/CVPR.2006.42.

[44] M. Oren et al. “Pedestrian Detection using Wavelet Templates”. In: *CVPR*. 1997. DOI: 10.1109/CVPR.1997.609319.

[45] O. M. Parkhi et al. “Cats and Dogs”. In: *CVPR*. 2012. DOI: 10.1109/CVPR.2012.6248092.

[46] D. Ramanan. “Learning to Parse Images of Articulated Bodies”. In: *NIPS*. 2006.

[47] S. Sarkar et al. "The HumanID Gait Challenge Problem: Data Sets, Performance, and Analysis." In: *PAMI*. Vol. 27. 2. Mar. 2005, pp. 162–77. DOI: 10.1109/TPAMI.2005.39.

[48] G. Shakhnarovich, P. Viola, and B. Moghaddam. "A Unified Learning Framework for Real Time Face Detection and Classification". In: *Automatic Face and Gesture Recognition*. 2002. DOI: 10.1109/AFGR.2002.1004124.

[49] B. Singh et al. "A Multi-stream Bi-directional Recurrent Neural Network for Fine-Grained Action Detection". In: *CVPR*. 2016. DOI: 10.1109/CVPR.2016.216.

[50] M. Stark et al. "Fine-Grained Categorization for 3D Scene Understanding". In: *BMVC*. 2012. DOI: 10.5244/C.26.36.

[51] C. Szegedy et al. "Going Deeper with Convolutions". In: *CVPR*. 2015. DOI: 10.1109/ICCV.2011.6126456.

[52] C. Wah et al. *The Caltech-UCSD Birds-200-2011 Dataset*. Tech. rep. California Institute of Technology, 2011.

[53] J. Wang, K. Markert, and M. Everingham. "Learning Models for Object Recognition from Natural Language Descriptions". In: *BMVC*. 2009. DOI: 10.5244/C.23.2.

[54] P. Welinder et al. "The Multidimensional Wisdom of Crowds". In: *NIPS*. 2010.

[55] Y. Yang and D. Ramanan. "Articulated Pose Estimation using Flexible Mixtures of Parts". In: *CVPR*. 2011. DOI: 10.1109/CVPR.2011.5995741.

[56] N. Zhang et al. "PANDA : Pose Aligned Networks for Deep Attribute Modeling". In: *CVPR*. 2014. DOI: 10.1109/CVPR.2014.212.

Chapter 5

PERSON REIDENTIFICATION

The contents of this chapter are from the unpublished work ”Putting Pose into Person Reidentification” by D. Hall and P. Perona. It is in submission at ICCV 2017 at the time of publication of this thesis.

5.1 Abstract

For deep neural network based person reidentification, a network needs to learn a representation of a person that is invariant to the pose of that person. The idea that pose information should be explicitly used to make learning this invariant representation easier for a network is introduced. Pose is utilised by extracting patches around body part locations that have been estimated using state-of-the-art pose estimation methods. These patches are used as inputs to weight sharing convolutional neural networks that output a feature vector representing the similarity of two part patches. To aggregate the similarity features generated for each part into a single representation, two different strategies are proposed. Experiments show that rank-1 matching rates increase by 22% on CRP and 25.6% on hand labelled CUHK03 when compared to other deep neural network methods.

5.2 Introduction

Reidentifying individuals across fixed cameras with non-overlapping views is important for surveillance systems in airports, train stations and cities. It reduces the time it takes to recognise suspicious behaviour as well as to identify perpetrators of a crime. Recognising people from a moving camera that views pedestrians at different locations as well as across large time scales has applications in robotics and automated vehicles.

The person reidentification task is best described as the problem of finding a match for a given query image from a gallery of candidate images. This is a challenging task since images of the same individual can vary significantly due to changes in clothing, viewpoint, pose, lighting and background. The size of the gallery and the similarity between different individuals further impact the difficulty of the task (see Figure 5.1).

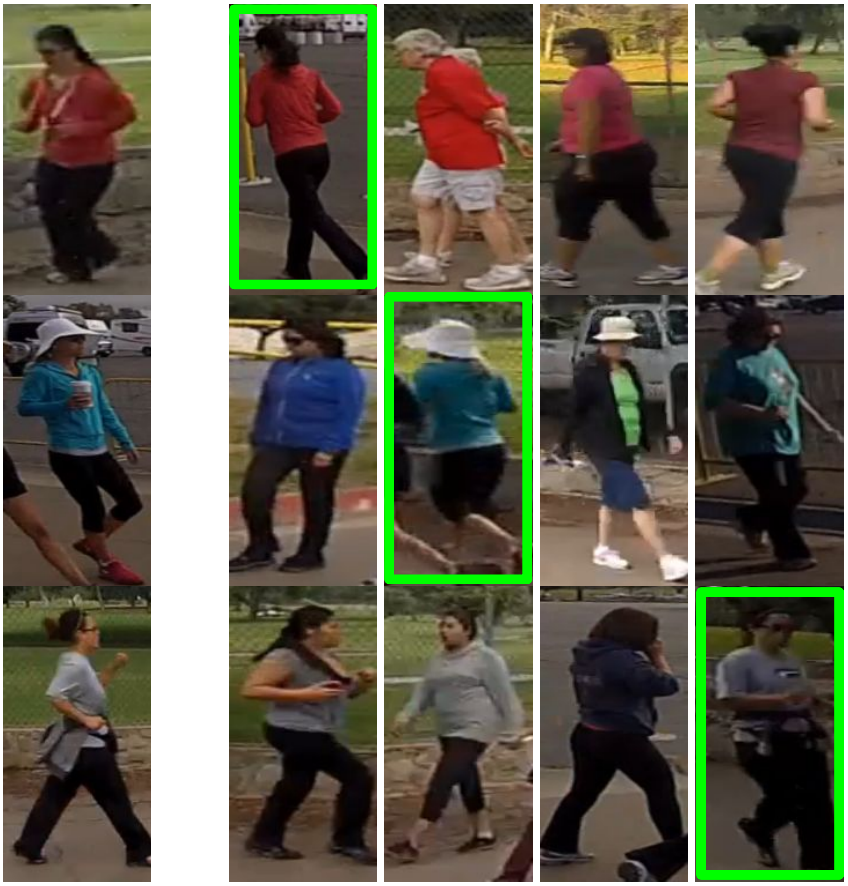


Figure 5.1: The person reidentification task. The left column contains examples of query images. On the right is the gallery of candidate images within which it is necessary to find a match for the query. The true match for each query is highlighted in green. Finding a match is a challenging task since the correct match can vary significantly from the query due to changes in clothing, viewpoint, pose, lighting and background. The similarity between different individuals also adds to the challenge.

Traditionally, person reidentification work has focused on engineering visual appearance features [17, 13, 30, 49, 42] and the design of metric learning algorithms[17, 39, 30, 49, 20, 19, 23, 27, 29, 42, 45]. More recently, deep convolutional neural networks (DCNN's) have been used to *learn* features and a similarity function jointly [22, 44, 2, 36].

One can imagine that for person reidentification to be successful, a network would need to learn a representation of a person that is invariant to the pose of that person. This representation may encode that the person is wearing a white hat, blue shorts, black sunglasses and a red shirt, but most importantly, the network needs to have the same representation whether the person is front on, side on or viewed from behind.

Existing deep learning approaches [22, 44, 2, 36] use the whole image of a person and expect that pose invariance is implicitly learnt by the network. However, this seems like a challenging task, particularly if the network is never told anything about pose.

To make learning a pose invariant representation easier for the network, an approach that explicitly uses pose information is introduced. Rather than use the whole image of a person as an input to the network, patches are extracted around body part locations. To determine the location of the body parts, state-of-the-art pose estimation techniques are used.

A two-stage deep neural architecture is proposed. The first stage is a set of weight sharing similarity networks that compare two patches of a particular body part from two different people. This stage outputs a similarity feature vector. The second stage is a feature aggregation network that combines the similarity feature vectors for each part to produce a single representation to be input into a classifier.

In addition to this we also introduce a new dataset for person reidentification. Traditionally, person re-identification has been studied from a surveillance paradigm where pedestrians are matched across fixed cameras with non-overlapping views. However, with the increasing popularity of automated vehicles and with a number of companies collecting street-view imagery (Google, Carmera, Mapillary), it is important to consider the person reidentification problem in the moving-camera domain. We add identity labels to an existing roadside pedestrian dataset CRP [6]. These labels will be made publicly available.

To summarise, we make three main contributions:

1. That pose information should be utilised to make learning a pose invariant representation for person reidentification, easier for convolutional neural networks.
2. A two-stage deep neural network architecture that uses body part patches as inputs rather than an entire image of a person.
3. The addition of identity labels to an existing roadside pedestrian dataset [6], to study person reidentification in the moving-camera domain.

5.3 Related work

Typically, person reidentification work focuses on engineering features that are quasi-invariant to changes in illumination, background and pose. Features that have been proposed include colour histograms [17, 13, 30, 49, 42], Gabor and Schmid filters [17, 30, 49], image patches [47, 46] and semantic colour names [15]. To incorporate spatial information, features are usually computed on horizontal strips of a pedestrian image [17, 30, 49, 25]

Another major area of focus is metric learning [17, 39, 30, 49, 20, 19, 23, 27, 29, 42, 45]. Given a set of features, the goal is to learn a similarity function that forces features of matching image pairs to be closer than those of non-matching images.

With the recent success of deep convolutional neural networks across many problems in computer vision [21, 7, 32, 33, 16], there has been an effort to move person reidentification away from hand designed features. There have been a number of recent works that use deep learning for person reidentification [22, 44, 2, 8, 37, 36, 38, 41]

Li *et al*. [22] were the first to use deep learning for person reidentification with the *filter pairing neural network* (FPNN). The network has six layers and it takes two images as input (at both train and test time) and outputs the probability that they match. The internal layers are designed to handle misalignment, cross-view photometric and geometric transforms of the image pairs.

Yi *et al*. [44] introduce siamese convolutional networks for person reidentification. Their model contains three separate siamese networks, each being independently trained on one of three sections of the two input images (the images are split into head, body and leg sections). The outputs of each siamese network are fused to produce a final similarity score.

The work of Ahmed *et al*. [2] introduces a novel, cross neighbourhood input difference layer that compares the features from one input image with the features computed in neighbouring locations of the other image. It currently has state-of-the-art performance on the CUHK03 [22] labelled dataset.

More recent work includes that of Cheng *et al*. [8] who use a multi channel CNN framework on global and local features. They refer to the local features as ‘body part’ features; however, these do not correspond to semantic body parts but are a partitioning of the global convolution layer.

Adding pose information into models has proven useful for a number of computer vision problems. Branson *et al*. [5] introduced ’pose normalised deep convolutional nets’ for the fine-grained categorisation of bird species. In this framework, given an image, keypoint part locations of a bird are predicted (beak, leg, wing *etc*.). The parts are then aligned to a set of prototypical models. Features are then extracted from the pose normalised image using deep nets. It has state-of the-art performance on bird species recognition. There is also the work of Taigman *et al*. [33] where pose is used to align faces to a 3D model. This improved face-verification results by 27% from prior state-of-the-art.

The work of Wu *et al*. [40] and Garcia *et al*. [14] use viewpoint based pose in the person reidentification domain where each image is assigned a viewing angle (a single scalar). This differs from the part-based or articulated pose approach we are proposing where (x,y) locations are assigned to multiple semantic body part locations.

There is a great deal of work on human pose estimation and a full discussion of the topic is outside the scope of this paper. We refer the reader to [28] for a summary of the field.

5.4 Approach

A person reidentification algorithm takes as input a query and a candidate image. It outputs a score indicating how well the two images match. A true match occurs when the query and candidate have the exact same person in them. This is repeated for every candidate image. The candidates are then sorted based on the score they received.

In this section the details of our algorithm are outlined. Firstly, given a query and a candidate image we explain how part patches are extracted. Secondly, for a single part patch, we describe the convolutional network used to produce a feature vector that encodes how similar the query and candidate patches are. Finally, we detail how the similarity features for each part are combined to form an aggregated feature vector which is then used by a final classification layer. A diagram of our pipeline can be found in Figure 5.2

Part Extraction

To extract patches from an image at body part locations, we must first use a pose estimation algorithm [43, 7, 35, 34, 28] to estimate where these body parts are. Fortunately, human pose estimation has greatly improved in recent years with state-of-the art methods being fast and accurate.

In this work we use the hourglass network for pose estimation [28] which has state-of-the-art performance on MPII [3] (a pose estimation dataset). Patches are

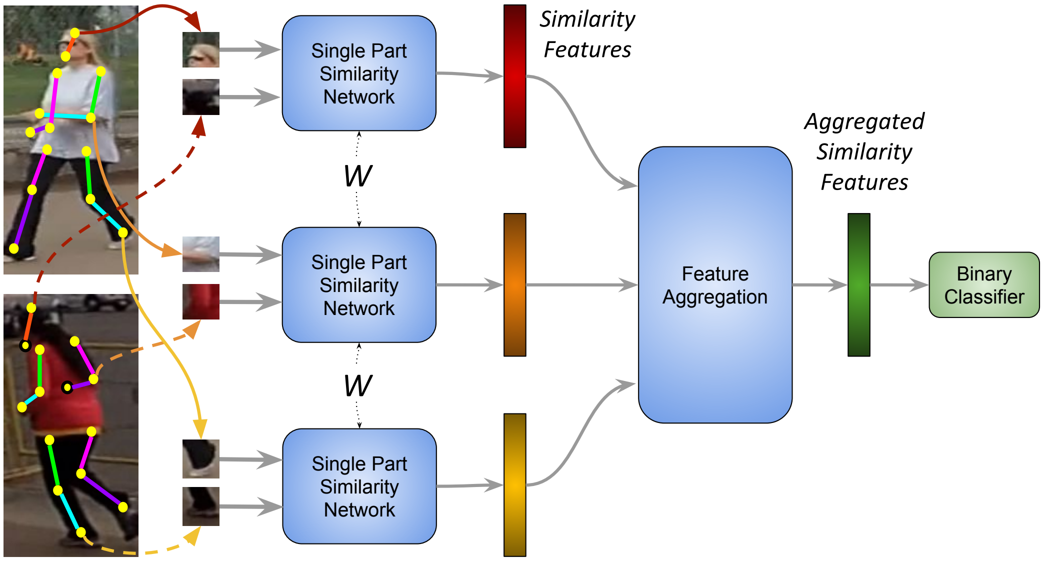

Single Part Similarity Network
W
Single Part Similarity Network
W
Single Part Similarity Network
Similarity Features
Feature Aggregation
Aggregated Similarity Features
Binary Classifier

Figure 5.2: The Proposed Person Reidentification Pipeline. Body part locations are estimated by an ‘as is’ pose estimator [28] for a query and a candidate image. 30 × 30 part patches are then extracted from the 128 × 64 images and fed into the single part similarity network which produce similarity features. The similarity features for each part are then aggregated and a final classification layer determines whether the query and candidate have the exact same person in them. The similarity networks have the same set of parameters. We show the case for three parts but in practice all of the sixteen parts that the pose estimator predicts are used.

extracted around the 16 body part locations that the pose estimator has been trained to find (top of the head, chin, chest, groin, shoulders, elbows, wrists, hips, knees and ankles). Each part patch is a 30 × 30 crop centred at the (x, y) keypoint location. The crop is extracted from a 128 × 64 image.

Single Part Similarity Network

Given a patch from the query and a patch from a candidate for a particular body part (*e.g.* the left ankle), a similarity network is used to determine whether these patches correspond to the same person.

The similarity network has the following architecture. There are two convolutional neural networks (CNN) that share the same parameters (they have identical weights and biases). The first CNN takes the query patch as input and the second takes the candidate patch. The size of the inputs are 30 × 30 × 3. The CNNs have two repeated blocks. Each block contains a convolutional layer with a 3x3 kernel and 25 channels followed by a ReLU [21] activation function and then a max-pooling layer which have 2x2 kernels with no-overlap. The CNNs each output a vector that

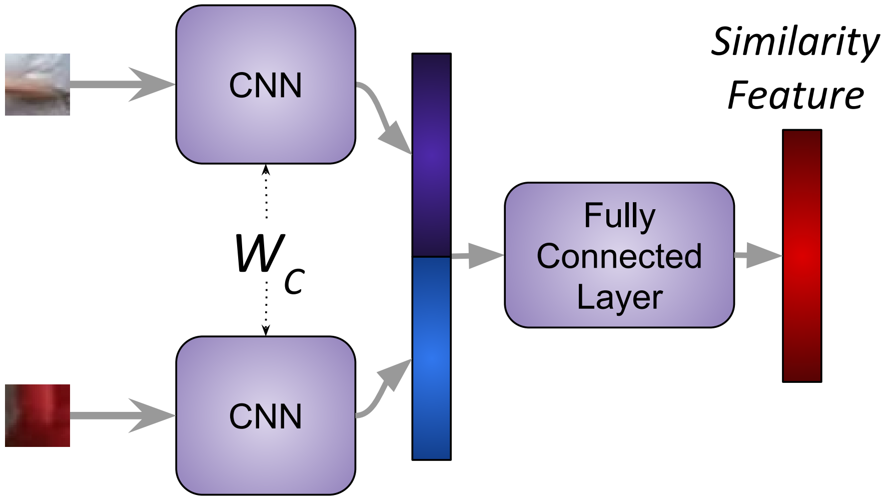


Figure 5.3: Single Part Similarity Network. The similarity network takes two 30 × 30 × 3 part patches as inputs to two convolutional neural networks that have the same parameters. Each CNN has a 3 × 3 × 25 convolutional layer followed by a max pooling layer. This structure is repeated twice. ReLU activations are used. Each CNN outputs a 1600 × 1 vector which are concatenated. A final fully-connected layer is then used to output a similarity feature vector of 500 dimensions.

has size 1600 × 1. These two outputs are then concatenated and passed through a fully-connected layer to produce a similarity feature vector of 500 dimensions. A diagram of the architecture can be found in Figure 5.3.

Since 16 body part locations are being used 16 similarity networks are required. A drawback of this formulation is that the number of parameters in the entire network grows with the number of body parts. To avoid this we also have each similarity network share parameters.

Feature Aggregation

The next question to ask is how are the similarity features for each of the 16 parts aggregated into a single representation? Two possibilities are considered for feature aggregation: 1) a basic approach where each of the 16 similarity feature vectors are concatenated. 2) a more sophisticated approach using an L layer, 16 time step recurrent neural network (RNN). At each time step a similarity feature vector for a particular part is input into the RNN. The aggregated feature vector is the output of the RNN at the final step. We use Gated Rectified Units [10] (GRUs) for our recurrent neural networks with 200 dimensions. In our experiments we consider the $L = 1$ and $L = 2$ cases. A diagram of each of the Feature Aggregation strategies can be found in Figure 5.4

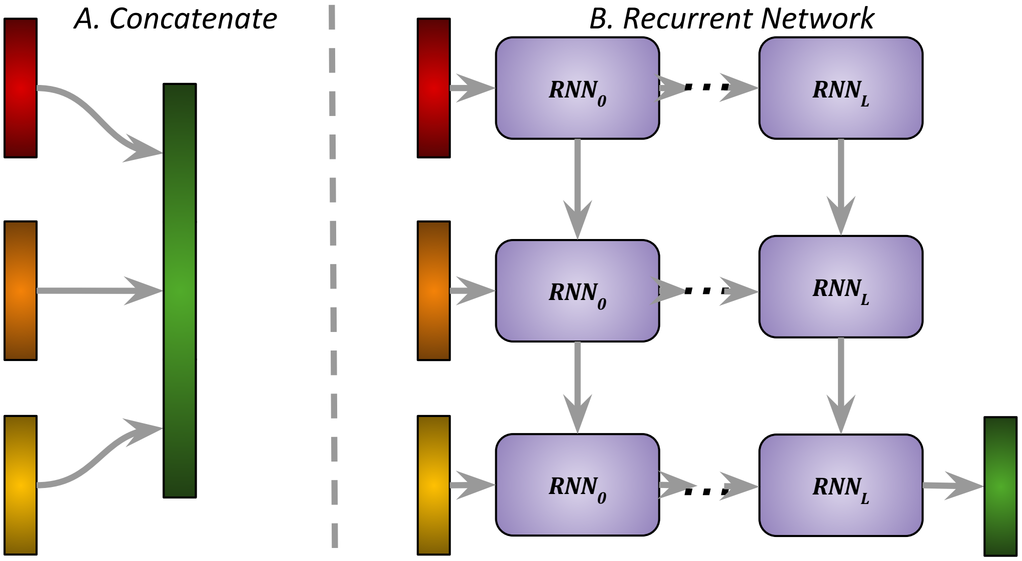


Figure 5.4: Feature Aggregation Strategies. Two possibilities are considered for feature aggregation: A) each of the 16 similarity vectors are concatenated. B) An L layer, 16 time step recurrent neural network that takes a similarity feature vector as an input at each time step and outputs the aggregated feature vector at the final step. We use Gated Rectified Units [10] (GRUs) for our recurrent neural networks with 200 dimensions. In our experiments we consider the $L = 1$ and $L = 2$ cases

The aggregated feature vector is then passed through a final fully-connected classification layer. The classification layer outputs a one if the query and candidate have the exact same person in them and zero otherwise. The cross-entropy loss is used during training.

5.5 Datasets

In this work we conduct experiments on 1) CUHK03 [22], an existing dataset with a fixed-camera paradigm and 2) Caltech Roadside Pedestrians (CRP) [6], an existing moving-camera dataset for which we have collected identity labels. As mentioned in Section 5.2 with the increasing prevalence of automated vehicles, it is important to consider the person reidentification problem in the moving-camera domain. To the best of our knowledge, we are the first to conduct person reidentification experiments in the moving-camera framework.

While there are other standard person reidentification datasets [17, 31, 48, 9], they are all quite small, making it difficult to train deep networks with them. However, both CRP and CUHK03 have a large number of examples, making them appropriate for studying CNNs.

CUHK03

The CUHK03 [22] dataset contains 13,158 images of 1,359 unique pedestrians across 2,934 sequences. Each individual occurs in exactly 2 sequences. Each sequence contains on average 4.8 pedestrian images. The dataset is captured using ten surveillance cameras with disjoint views. They monitor an open area where pedestrians walk in different directions. This dataset doesn't contain any ground truth pose labelling. It contains hand labelled pedestrian bounding boxes as well as boxes automatically detected by a deformable-part-model. We include results for both cases.

CRP

CRP [6] is a large video dataset of pedestrians. It contains 30,000 bounding box annotations of pedestrians in 4222 tracks (sequences) across 7 videos collected on different days and times. For each bounding box, there are 14 body parts labelled. CRP was collected by mounting a camera to a vehicle which did laps of a park [6]. Two features of this dataset are as follow: 1) The camera had a fish-eye lens so within a track the same person is captured from all viewpoints (so front, side, back). 2) Since the vehicle is doing laps, individuals are seen again at different points in time, in different locations, with different backgrounds, different lighting conditions and sometimes even different clothing. For further details about the dataset refer to Chapter 4.

When CRP [6] was originally collected it included many types of annotations but **it lacks global identity labels** for the pedestrian sequences. To make it possible to study person reidentification from a moving-camera using this dataset **we add to the annotations** by using Amazon Mechanical Turk to collect identity labels. Workers were given a probe pedestrian sequence and a gallery of 50 sequences. They were then asked to click on any sequence in the gallery that matched the probe. This resulted in us collecting labels for 1,155 unique individuals. Each individual occurs on average in 2.6 sequences. Each sequence contains on average 6.1 pedestrian images. An example of the interface can be found in Figure 5.10.

5.6 Experiments

Pose Estimation

To determine the location of body parts we use Newell *et al.*'s [28] hourglass network for pose estimation. It has state-of-the-art performance on the pose estimation dataset

MPII [3]. In all of our experiments we use the publicly available model[1] provided by Newell [28]. This model has been trained on MPII. We use this model **as is**, without any fine-tuning on CRP or CUHK. This pose estimator predicts 16 body part locations.

Proposed Model Variants

We conduct experiments using the following model variants: (a) RNN feature aggregation with 2 layers **(2-RNN)**. (b) RNN feature aggregation with 1 layer **(1-RNN)**. (c) Concatenated feature aggregation **(Concat)**. (d) RNN feature aggregation with 2 layers but weights are *not shared* between the 16 similarity networks **(2-RNN+no_sharing)**. (e) RNN feature aggregation with 2 layers and the similarity feature vectors are input into the RNN in no particular order **(2-RNN+shuffled)**. Rather than input the similarity feature vectors in an arbitrary fixed order (the left ankle at time-step one, the left knee at time-step two. etc.) we experiment with inputting the feature vectors for a particular part at any time-step. For example, for the first pair of images the left ankle might be input at time-step 3 but for the second pair it might be input at time-step 14.

Each variant has the following number of parameters: 2-RNN has $1,893,251$; 1-RNN has $1,652,651$; Concat has $1,247,852$; and 2-RNN+no sharing has $20,232,001$. In comparison CIND [2] has $2,308,147$ parameters.

Baseline Models

We also consider two baseline approaches to compare against. The first is the **Whole Image** method. For this approach the **Concat** network architecture is used but instead of using 16, 30x30 patches as input, the entire image is used (a single input of 128x64).

The second is the **Low Level** method. In this approach, for each of the extracted part patches a colour histogram (8x8x8 HSV descriptor) and a Scale Invariant Local Ternary Pattern (SILTP) [24] descriptor is computed. XQDA [25] is then used to learn a metric from these descriptors. This approach is similar to that of [25] except that features are extracted from patches at the part locations rather than densely sampling the image in rows, then max pooling.

[1] https://github.com/anewell/pose-hg-demo

Data Splits

The CUHK03 dataset was split into a training set of 1160 identities, a validation set of 100 identities, as well as a test set of 100 identities. These sets come from cameras pairs 1-6. This is the standard set up as described in [22] and is the setting in which all prior work reports results on.

The CRP dataset had a train set of 5 videos containing 1636 individuals (870 occur more than once) in 14,769 images. A validation set of 1 video containing 260 individuals (130 occur more than once) in 2326 images. A test set of 1 video containing 209 individuals (117 occur more than once) in 2,039 images.

The validation sets for both datasets were used to select the network parameters.

Training Parameters

Our network is implemented using *keras* [11] with *tensorflow* [1] as a backend. We use the stochastic optimisation method, Adam [4] for training. Each mini-batch has 20 pairs with an equal ratio of matching image pairs to non-matching image pairs (so 10 of each). The identity of the pairs as well as the image for each selected identity is chosen at random.

We train for 200 epochs with a learning rate of 0.0001. The learning rate is then reduced to 0.00001 and the network is trained for a further 20 epochs. There are $20,000$ training examples per epoch. Weight decay is also used for regularisation with a value of 0.0005.

To avoid over-fitting we also augment the part patches extracted from an image. Each patch undergoes a random translation of up to 5% of its height and width, a random rotation of up to 10° and a random horizontal flip.

CUHK03

The CUHK03 experiment we consider is the standard test case. In this setting there are 100 query sets. For each query set a query image is selected at random. There is also a set of 100 candidate sets. A candidate image is selected at random from each candidate set to form a gallery of 100 images. There is **only one** matching image in the gallery. The gallery images are then ranked according to how well their identities match the query. This is repeated 10 times and the average cumulative-match-characteristic (CMC) is reported.

We compare against several existing methods:

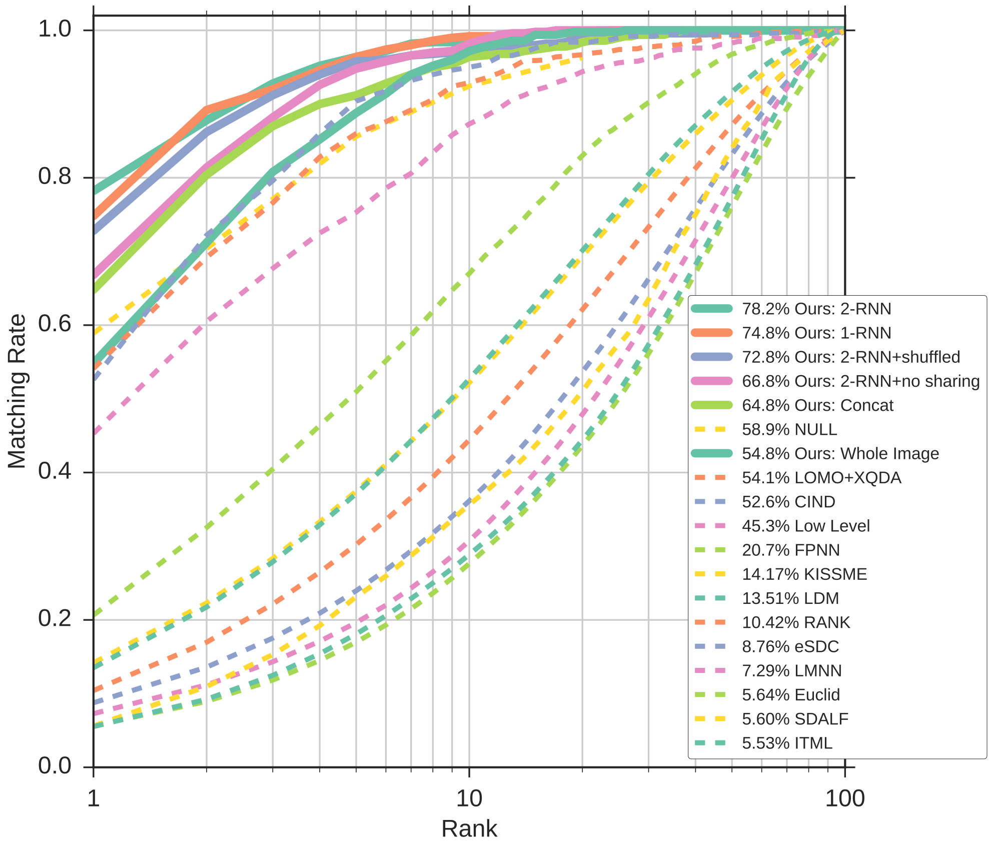


Figure 5.5: The CMC for labelled CUHK03. The **2-RNN** variant is 25.6% better at rank-1 than CIND [2], the state-of-the-art deep learning approach. It is also 19.3% better than NULL [45] at rank-1, a metric learning method that operates on top of pre-computed LOMO [25] features. Solid lines correspond to variants of our proposed method. Dashed lines correspond to state-of-the-art methods.

(a) Older, metric learning based methods: KISSME [20], eSDC [47], SDALF [13], ITML [12], LDM [18], LMNN [39], RANK [26], and the euclidean distance. These methods use dense colour histograms and dense SIFT for features [22].

(b) More recent metric learning methods NULL [45] and XDQA [25] which both use LOMO [25] features.

(c) Deep learning methods FPNN [22], CIND [2] (which we have implemented ourselves since it has state of the art performance) and GATED [36] (only reports results on detected).

Results on hand labelled images are show in Figure 5.5. Our **2-RNN** variant is 25.6% better at rank-1 than CIND, the best existing deep learning approach. It is 19.3% better than NULL at rank-1, a metric learning method that operates on top of

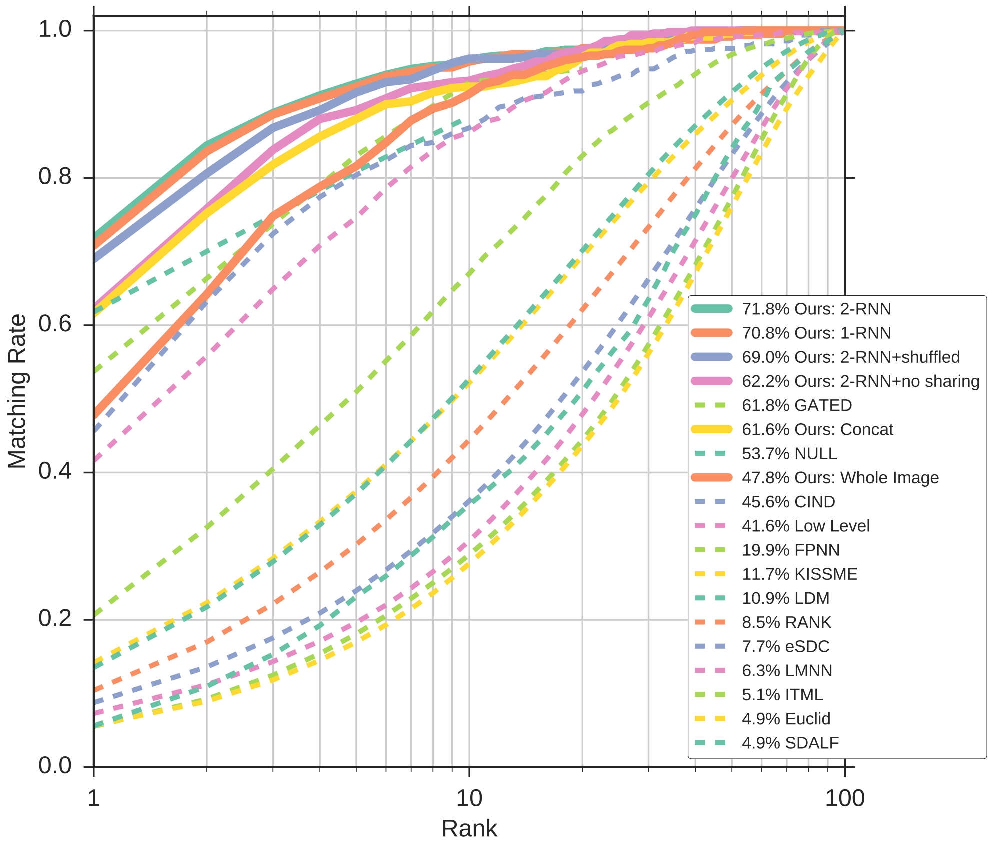


Figure 5.6: The CMC for detected CUHK03. We are doing 26.2% better than CIND and 10% better than GATED [36]. The **2-RNN** variant is the best performing variant, performing 13.4% better than **Concat**.

pre-computed LOMO [25] features.

The **2-RNN** variant is 3.4% better than the **1-RNN** variant and 13.4% better than the **Concat** variant at rank-1 with both these variants still improving on the state-of-the-art results. The **2-RNN+no sharing** variant performs 11.4% worse at rank-1 than its weight sharing counterpart probably due to over-fitting. When the inputs to the RNN are presented in no particular order a significant improvement is still seen over state-of-the-art.

The **Whole Image** baseline model does surprisingly well with a rank-1 performance close to both CIND and NULL. However, **Concat**, its part-based equivalent still performs 9.2% better. This indicates that explicitly using the part information helps the network to learn a better representation. LOMO+XQDA performs 8.8% better than its part-based equivalent, the **Low Level** baseline model. This result indicates that parts aren't as useful when a feature designing+metric learning approach is used

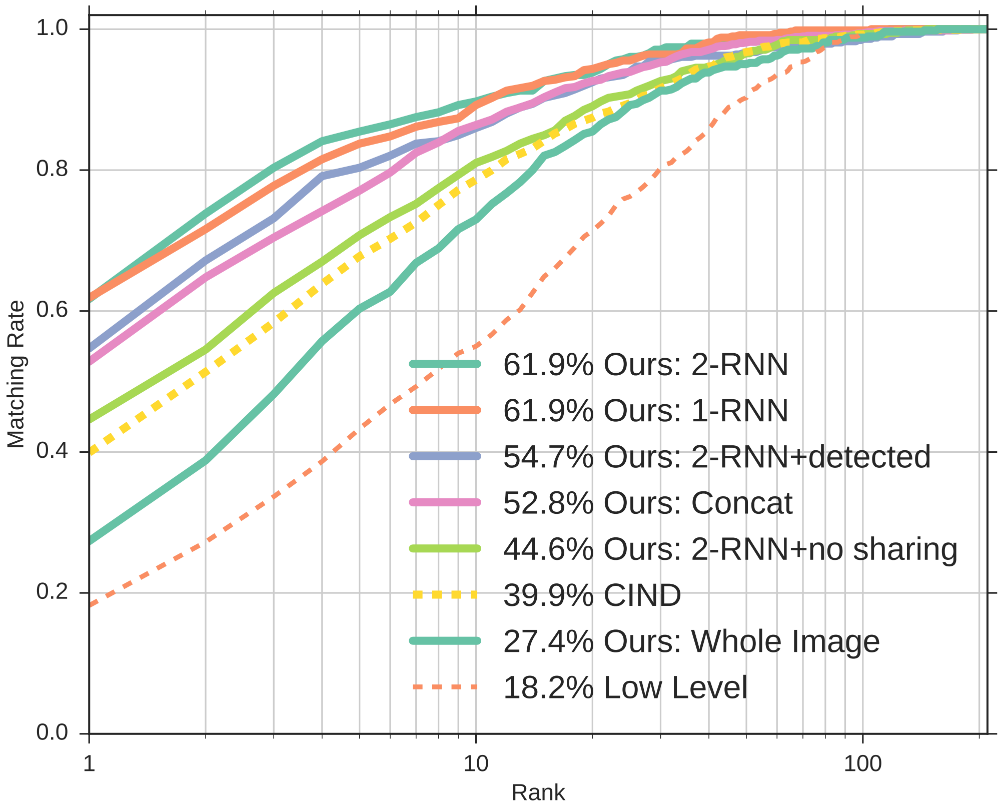


Figure 5.7: The CMC for CRP. The **2-RNN** variant is 22% better at rank-1 than CIND [2]. The **2-RNN** variant is again the best performing variant, performing 22% better than **Concat**.

compared to an end-to-end deep neural network method.

Similar results using detected images can be found in Figure 5.6. Again, the **2-RNN** variant is doing 26.2% better than CIND and 10% better than GATED.

CRP

We also conduct experiments on the CRP dataset and compute the average CMC. In this setting there are 117 query sets (since this is the number of individuals that occur more than once in the test set.) A query image is selected at random from each query set. There are 209 candidate sets. A random image is selected from each set to form a gallery of 209 images. There is **only one** matching image in the gallery. The gallery images are then ranked as before and the process repeated 10 times.

Results can be found in Figure 5.7. The **2-RNN** variant is 22% better at rank-1 than CIND. When comparing the different variants we see a similar ranking to the CUHK03 experiments with the results again demonstrating that the 2 layer RNN

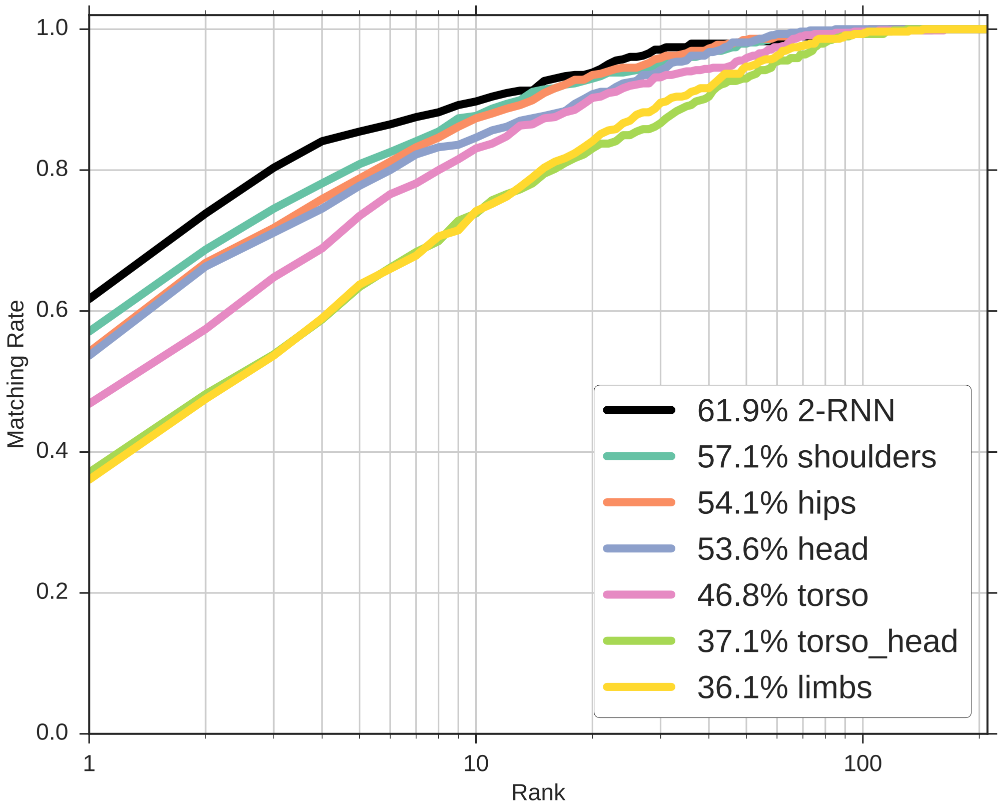


Figure 5.8: The Impact of Removing Different Parts. Instead of inputting all parts into the network, only a subset of parts are input. Parts are removed by setting the input patches to zero. The parts that were **removed** are: 1) the head (chin and top of head), 2) the left and right shoulders, 3) the left and right hips, 4) the torso (shoulders and hips), 5) the torso and the head, and 6) the limbs (all elbows, wrists, ankles and knees). The results suggest that the head and shoulders are less important than the limbs when discriminating individuals.

feature aggregation method is better than the concatenation method.

Impact of Removing Different Parts

In this section we run an experiment where instead of inputting all parts into the network, only a subset of parts are input. We remove parts by setting their 30×30×3 input patches to zero. We run this experiment to quantify the effect groupings of parts have on the CMC. Ground truth pose is used. The parts that we **remove** are: 1) the head (chin and top of head), 2) the left and right shoulders, 3) the left and right hips, 4) the torso (shoulders and hips), 5) the torso and the head, and 6) the limbs (all elbows, wrists, ankles and knees).

Results can be found in Figure 5.8. If only the hips or shoulders are removed there is a 4.8-7.8% reduction in performance when compared to using all available parts. This suggest that these parts only play a minor role in discriminating individuals from one another. On the other hand, if the limbs are removed, there is a 25.8% decrease in performance. This suggests that elbows, knees, wrists and ankles are important body parts for identity discrimination.

A reason for this could be that the shoulder patches give information about the overall colour of a person. When those patches are removed, the overall colour can still be identified in hip patches and possibly wrist or elbow patches. For the limbs, knee patches give information about whether someone has shorts or pants on, elbow patches whether there is long sleeves or short sleeves. If these patches are removed this information is harder to recover from other parts and so performance degrades.

Qualitative Comparison

Qualitative results, comparing the rankings produced by CIND [2] and our method can be found in Figure 5.9.

The top three rows are results from CRP. Each of the three query images have a person wearing a white hat. If we look at the rankings returned by CIND we notice that the rank-1 images have a feature that is similar to the query despite them all being an incorrect match. In row one they have white tops, in row two both have a cap, and in row three both have a white hat. Lower ranked images tend to be matching in overall colour, such as in row two, where everyone has dark clothes. The rankings returned by our method are more consistent, in that the highest ranked images all have a white hat. This is one of the reasons why our method has a better CMC.

The bottom three rows are results from CUHK03. The rank-1 matches using the CIND method are what we call ‘unreasonable errors’ since the query image is visually, completely different. In row four, the query has a person with a grey shirt and blue pants but the rank-1 image is a person with a purple shirt and shorts. In row five, the query has dark clothing while the best match has a red backpack. In row six, the query has a light, patterned shirt while the top match has a dark shirt. The results returned by our method do not have as many ‘unreasonable errors’. Rows five and six are the correct match. While for row one, even though the correct match is at rank-2, the top match has a light top and blue pants, which is visually similar to the query.

5.7 Discussion and Conclusions

Existing deep learning approaches [22, 44, 2, 36] use the whole image of a person and expect that pose invariance is implicitly learnt by the network. However, this is a challenging task, particularly if the network is never told anything about pose.

We claim that pose information should be utilised to make learning a pose invariant representation for person reidentification easier for convolutional neural networks. To the best of our knowledge, using part-based pose has never been used before in the person reidentification setting.

We utilise the pose by extracting patches around body part locations that have been estimated using a state-of-the-art pose estimator [28]. The pose estimator is used **as is** with no fine-tuning on any person reidentification dataset. These patches are used as inputs to weight sharing convolutional neural networks that output a feature vector representing the similarity of two part patches. To aggregate the similarity features generated for each part into a single representation, a number of different variants were considered. The best performing feature aggregator was the two-layer recurrent neural network.

We also add identity labels to an existing roadside pedestrian dataset [6], to study person reidentification in the moving-camera domain. These labels will be made publicly available.

Experiments show that by using our approach, rank-1 matching rates increase by 22% on CRP. On hand labelled CUHK03 there is an increase of 25.6% when compared to other, deep neural network methods and a 19.3% increase when compared to the best metric learning method. Similar increases are observed on detected CUHK03.

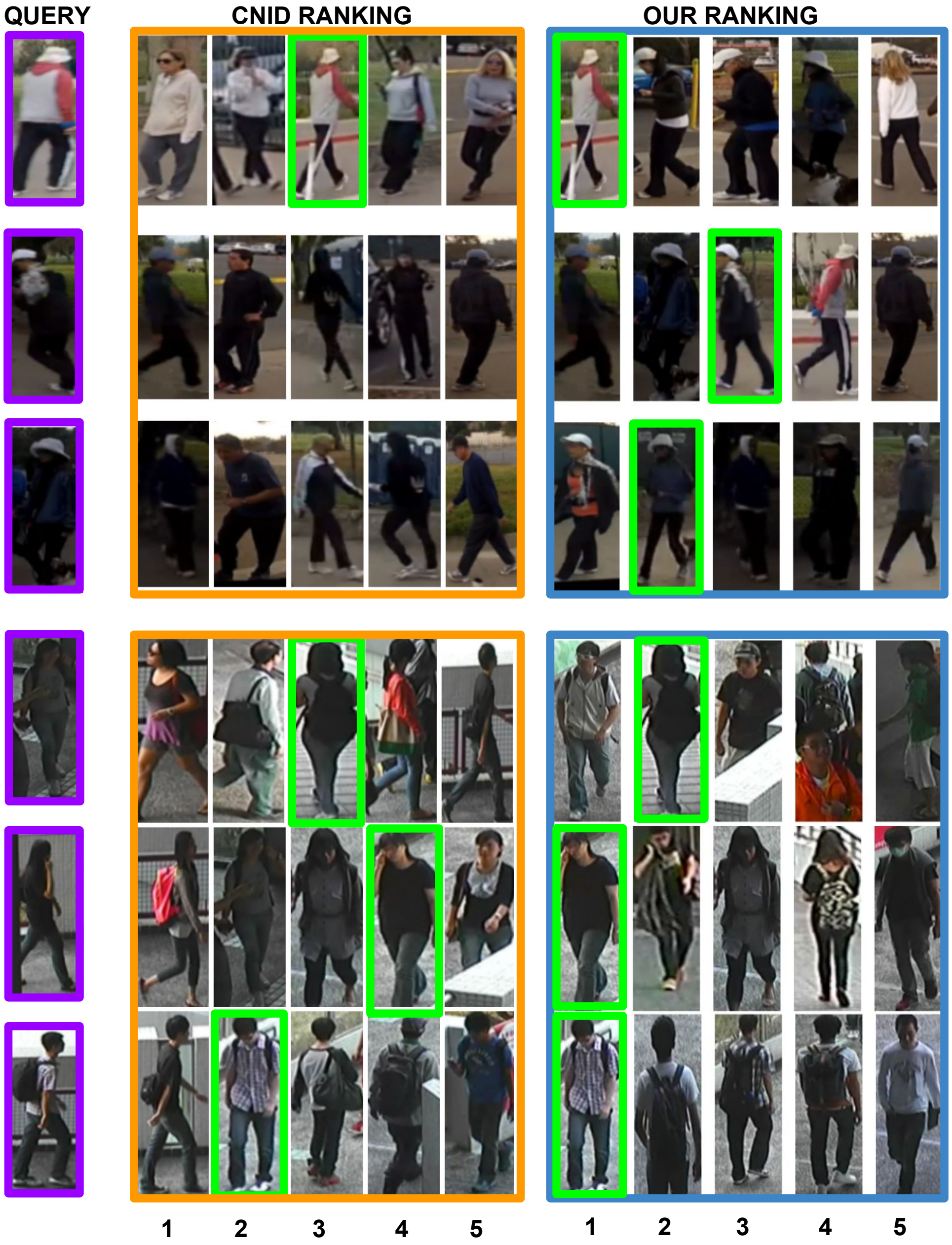
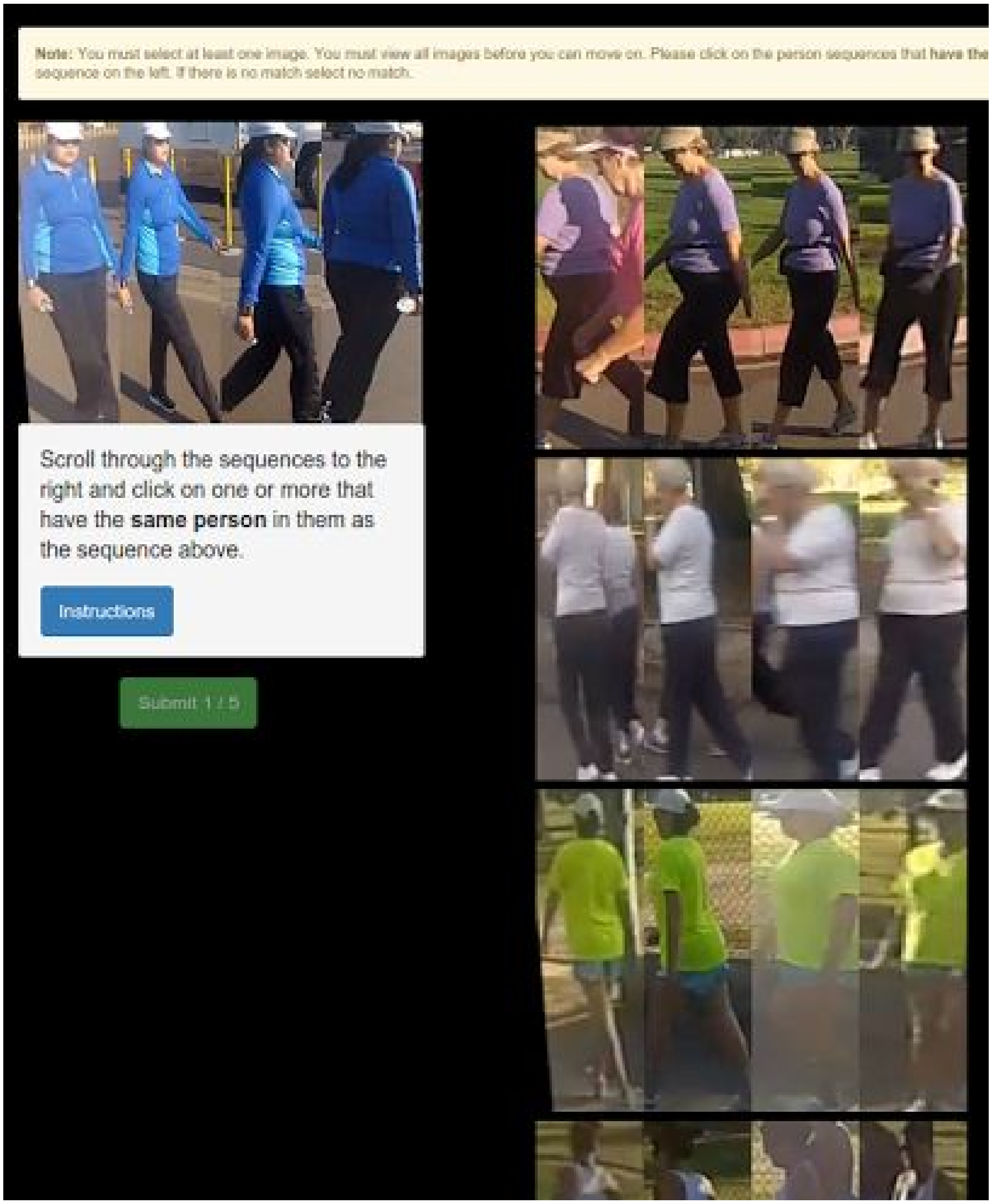


Figure 5.9: Example Ranking Results. Qualitative results, comparing the rankings produced by CIND [2] (yellow) and our method (blue). The top three rows are results from the CRP dataset. The bottom three are from CUHK03. Query images have purple borders and the correct matches in the candidate sets have green borders. For a discussion of these results see Sec 5.6

5.8 Appendix: Dataset Collection



Note: You must select at least one image. You must view all images before you can move on. Please click on the person sequences that **have the** sequence on the left. If there is no match select no match.

Scroll through the sequences to the right and click on one or more that have the **same person** in them as the sequence above.

Figure 5.10: The interface used by Amazon Mechanical Turk workers to label the identity of a person in the Caltech Roadside Pedestrians Dataset. The user is presented with an example individual on the left. They then scroll through the examples on the right, clicking on the ones that match.

References

[1] M. Abadi. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. 2015.

[2] E. Ahmed, M. Jones, and T. K. Marks. "An Improved Deep Learning Architecture for Person Re-Identification". In: *CVPR*. 2015. DOI: 10.1109/CVPR.2015.7299016.

[3] M. Andriluka et al. "2D Human Pose Estimation: New Benchmark and State of the Art Analysis". In: *CVPR*. 2014. DOI: 10.1109/CVPR.2014.471.

[4] J. Ba and D. Kingma. "Adam: A Method for Stochastic Optimization". In: *ICLR*. 2015.

[5] S. Branson et al. "Improved Bird Species Recognition Using Pose Normalized Deep Convolutional Nets". In: *BMVC*. 2014. DOI: 10.5244/C.28.87.

[6] D. Hall and P. Perona. "Fine-Grained Classification of Pedestrians in Video: Benchmark and State of the Art". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015. DOI: 10.1109/cvpr.2015.7299187.

[7] X. Chen and A. Yuille. "Articulated Pose Estimation by a Graphical Model with Image Dependent Pairwise Relations". In: *NIPS*. 2014. DOI: 10.1.1.672.1960.

[8] D. Cheng et al. "Person Re-Identification by An Multi-Channel Parts-Based CNN with Improved Triplet Loss Function". In: *CVPR*. 2016. DOI: 10.1109/CVPR.2016.149.

[9] D. S. Cheng et al. "Custom Pictorial Structures for Re-identification". In: *BMVC*. 2011. DOI: 10.5244/C.25.68.

[10] K. Cho et al. "Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation". In: *EMNLP*. 2014.

[11] F. Chollet et al. *Keras*. https://github.com/fchollet/keras. 2015.

[12] J. V. Davis et al. "Information-Theoretic Metric Learning". In: *ICML*. 2007. DOI: 10.1145/1273496.1273523.

[13] M. Farenzena et al. "Person Re-Identification by Symmetry-Driven Accumulation of Local Features". In: *CVPR*. 2010. DOI: 10.1109/CVPR.2010.5539926.

[14] J. Garcia et al. "Modeling Feature Distances by Orientation Driven Classifiers for Person Re-identification". In: *JVCIR* 38 (2016), pp. 115–129. DOI: 10.1016/j.jvcir.2016.02.009.

[15] N. Gheissari et al. "Person Reidentification using Spatiotemporal Appearance". In: *CVPR*. 2006. DOI: 10.1109/CVPR.2006.223.

[16] R. Girshick. "Fast R-CNN". In: *ICCV*. 2015. DOI: 10.1109/ICCV.2015.169.

[17] D. Gray and H. Tao. “Viewpoint Invariant Pedestrian Recognition with an Ensemble of Localized Features”. In: *ECCV*. 2008. DOI: 10.1007/978-3-540-88682-2_21.

[18] M. Guillaumin, J. Verbeek, and C. Schmid. “Is That You? Metric Learning Approaches for Face Identification”. In: *CVPR*. 2009. DOI: 10.1109/ICCV.2009.5459197.

[19] M. Hirzer et al. “Relaxed Pairwise Learned Metric for Person Re-identification”. In: *ECCV*. 2012. DOI: 10.1007/978-3-642-33783-3.

[20] M. Kostinger et al. “Large Scale Metric Learning from Equivalence Constraints”. In: *CVPR*. 2012. DOI: 10.1109/CVPR.2012.6247939.

[21] A. Krizhevsky, I. Sutskever, and G. E. Hinton. “ImageNet Classification with Deep Convolutional Neural Networks”. In: *NIPS*. 2012.

[22] W. Li et al. “DeepReId: Deep Filter Pairing Neural Network for Person Re-Identification”. In: *CVPR*. 2014. DOI: 10.1109/CVPR.2014.27.

[23] Z. Li et al. “Learning Locally-adaptive Decision Functions for Person Verification”. In: *CVPR*. 2013. DOI: 10.1109/CVPR.2013.463.

[24] S. Liao et al. “Modeling Pixel Process with Scale Invariant Local Patterns for Background Subtraction in Complex Scenes”. In: *CVPR*. 2010. DOI: 10.1109/CVPR.2010.5539817.

[25] S. Liao et al. “Person Re-identification by Local Maximal Occurrence Representation and Metric Learning”. In: *CVPR*. 2015. DOI: 10.1109/CVPR.2015.7298832.

[26] B. Mcfee and G. Lanckriet. “Metric Learning to Rank”. In: *ICML*. 2010.

[27] A. Mignon and F. Jurie. “PCCA: A New Approach for Distance Learning from Sparse Pairwise Constraints”. In: *CVPR* (2012). DOI: 10.1109/CVPR.2012.6247987.

[28] A. Newell, K. Yang, and J. Deng. “Stacked Hourglass Networks for Human Pose Estimation”. In: *ECCV*. 2016. DOI: 10.1007/978-3-319-46484-8.

[29] S. Pedagadi et al. “Local Fisher Discriminant Analysis for Pedestrian Re-identification”. In: *CVPR*. 2013. DOI: 10.1109/CVPR.2013.426.

[30] B. Prosser et al. “Person Re-Identification by Support Vector Ranking”. In: *BMVC*. 2010. DOI: 10.5244/C.24.21.

[31] W. R. Schwartz and L. S. Davis. “Learning Discriminative Appearance-Based Models Using Partial Least Squares”. In: *Brazilian Symposium on Computer Graphics and Image Processing*. 2009. DOI: 10.1109/SIBGRAPI.2009.42.

[32] C. Szegedy et al. “Going Deeper with Convolutions”. In: *CVPR*. 2015. DOI: 10.1109/ICCV.2011.6126456.

[33] Y. Taigman et al. “DeepFace: Closing the Gap to Human-Level Performance in Face Verification”. In: *CVPR*. 2014. DOI: 10.1109/CVPR.2014.220.

[34] J. Tompson et al. “Joint Training of a Convolutional Network and a Graphical Model for Human Pose Estimation”. In: *NIPS*. 2014.

[35] A. Toshev and C. Szegedy. “DeepPose: Human Pose Estimation via Deep Neural Networks”. In: *CVPR*. 2014. DOI: 10.1109/CVPR.2014.214.

[36] R. R. Varior, M. Haloi, and G. Wang. “Gated Siamese Convolutional Neural Network Architecture for Human Re-Identification”. In: *ECCV*. 2016. DOI: 10.1007/978-3-319-46448-0.

[37] R. R. Varior et al. “A Siamese Long Short-Term Memory Architecture for Human Re-Identification”. In: *ECCV*. 2016. DOI: 10.1007/978-3-319-46478-7.

[38] F. Wang et al. “Joint Learning of Single-image and Cross-image Representations for Person Re-identification”. In: *CVPR*. 2016. DOI: 10.1109/CVPR.2016.144.

[39] K. Q. Weinberger and L. K. Saul. “Distance Metric Learning for Large Margin Nearest Neighbor Classification”. In: *Journal of Machine Learning Research* 10 (2009), pp. 207–244. DOI: 10.1126/science.277.5323.215.

[40] Z. Wu, Y. Li, and R. J. Radke. “Viewpoint Invariant Human Re-identification in Camera Networks using Pose Priors and Subject-discriminative Features”. In: *PAMI* 37.5 (2015), pp. 1095–1108. DOI: 10.1109/TPAMI.2014.2360373.

[41] T. Xiao et al. “Learning Deep Feature Representations with Domain Guided Dropout for Person Re-identification”. In: *CVPR*. 2016. DOI: 10.1109/CVPR.2016.140.

[42] F. Xiong et al. “Person Re-Identification using Kernel-Based Metric Learning Methods”. In: *ECCV*. 2014. DOI: 10.1007/978-3-319-10584-0_1.

[43] Y. Yang and D. Ramanan. “Articulated Pose Estimation using Flexible Mixtures of Parts”. In: *CVPR*. 2011. DOI: 10.1109/CVPR.2011.5995741.

[44] D. Yi et al. “Deep Metric Learning for Person Re-identification”. In: *ICPR*. 2014. DOI: 10.1109/ICPR.2014.16.

[45] L. Zhang, T. Xiang, and S. Gong. “Learning a Discriminative Null Space for Person Re-identification”. In: *CVPR*. 2016. DOI: 10.1109/CVPR.2016.139.

[46] R. Zhao, W. Ouyang, and X. Wang. “Learning Mid-level Filters for Person Re-identification”. In: *CVPR*. 2014. DOI: 10.1109/CVPR.2014.26.

[47] R. Zhao, W. Ouyang, and X. Wang. “Unsupervised Salience Learning for Person Re-identification”. In: *CVPR*. 2013. DOI: 10.1109/CVPR.2013.460.

[48] W.-S. Zheng, S. Gong, and T. Xiang. “Associating Groups of People”. In: *BMVC*. 2009. DOI: 10.5244/C.23.23.

[49] W.-S. Zheng, S. Gong, and T. Xiang. “Person Re-Identification by Probabilistic Relative Distance Comparison”. In: *CVPR*. 2011. DOI: 10.1109/CVPR.2011.5995598.

Chapter 6

CONCLUSIONS AND FUTURE DIRECTIONS

In this thesis, I present work that advances the methods for machines to be visually aware of people with a focus on understanding who, along with their attributes, is where in video. Efforts are focused on improving algorithms in four areas of visual recognition: detection, tracking, fine-grained classification and person reidentification.

Each of these problems appear to be quite different on the surface; however, there are two broader questions that are answered across each of the works. The first, the machine is able to make better predictions when it has access to the extra information that is available in video. The second, that it is possible to learn on-the-fly from single examples. A summary of how each work contributes to answering these over-arching questions as well as its specific contributions to the problem domain are as follows:

In Chapter 2, a method for training detectors of individual objects from a boosted category detector was presented. Training happens in real-time using a single instance of an individual as a positive training example. The individual detectors make use of the category detector's feature computations; the thresholds for a single weak classifier are set using transfer learning. This ensures that the additional training and run-time costs for the individual detectors are minimal.

Experiments were carried out on two datasets containing faces and pedestrians. They suggest two conclusions: (a) both training and runtime computation of individual detectors is extremely inexpensive and, (b) the proposed method has both better tracking and reidentification performance than previous methods as well as operating at faster frame rates.

In Chapter 3, a novel tracking method which is designed to track objects belonging to a specific category was presented. The method makes use of a boosted category detector to identify target objects to track and of an individual-specific detector to track the target in subsequent video frames. The individual-specific detector is trained on-the-fly at almost no extra cost (using the method proposed in Chapter 2), making it possible for the tracker to operate in real-time. The well-known problem of drift is addressed by updating the individual-specific detector only when there are coincident category detections.

The performance of our method is compared to 9 state-of-the-art trackers and results show that it is as accurate as the most accurate competitor, but 20x faster. It is only slightly slower than the fastest competitor, but 10% more accurate. The experiments were carried out on two large (hundreds of thousands of detections), challenging and heterogeneous datasets of faces and pedestrians. This benchmark surpasses, both in method and set size, any such comparative evaluation in the literature.

In Chapter 4, a video dataset designed to study the fine-grained classification of people using the entire human body is introduced. Its novel and distinctive features are size, realism (natural behaviour, variety in viewpoint, moving camera), fine-grained multi-label attributes (sex, weight, clothing, age), detailed annotations, and public availability.

Two methods were proposed. The first is the deep neural network based unified model which takes a single image as input and outputs the class distributions for each of the four fine-grained categories in the CRP dataset. The unified model does only marginally better or about the same when compared to the baseline model for sex, age and style. For body type the unified model does 10% better than the baseline model. Compared to humans, classification performance is quite poor for age, body type and style (29.4-33.7% worse). Sex classification performance is better (only 15.4% worse).

The second method uses sequences of images as input. The results indicate that combining information from a sequence of images of an individual and then predicting the label is 3.5-7.1% better, in terms of class average accuracy, than independently predicting the label of each image, assuming that severely under-represented classes are ignored. A class is considered to be severely under-represented if it makes up less than 5% of the class labels. The best method for combining information is unclear from these experiments with all variants performing differently depending on the fine-grained category.

In Chapter 5, a method that uses pose information for person reidentification is proposed. Part-based pose has never been used before in the person reidentification setting. Pose is utilised by extracting patches around body part locations that have been estimated using some pose estimation technique. The patches are used as inputs to weight sharing convolutional neural networks that output a feature vector representing the similarity of two part patches. To aggregate the similarity features generated for each part into a single representation, a number of different variants were considered. The best performing feature aggregator was the two-layer recurrent neural network. Identity labels were also added to the dataset collected in Chapter 4 to study person reidentification in the moving-camera domain.

Experiments show that by using the proposed method, rank-1 matching rates increase by 22% on CRP. On hand labelled CUHK03 there is an increase of 25.6% when compared to other deep neural network methods and a 19.3% increase when compared to the best metric learning method. Similar increases are observed on detected CUHK03.

I'll finish this dissertation with a few thoughts on future directions for the problems of fine-grained classification and person reidentification. To do better on these problems the simple answer is to collect more data. Compared to the ImageNet dataset which is used for object detection/recognition, the datasets used for reidentification and fine-grained classification of people are tiny, being around one hundred times smaller. With deep neural networks transforming machine vision in the past five years, if these problems had larger training sets then performance would keep improving. However, I think the real challenge lies in being able to make predictions from a small number of training examples. Humans have the ability to learn new classes of object from very few examples because they are able to transfer what they know about other objects to the new example. Exploring better methods for transfer learning in machines is a direction I would suggest those interested in this field should take.