

A Thesis Submitted for the Degree of PhD at the University of Warwick

Permanent WRAP URL:

<http://wrap.warwick.ac.uk/144567>

Copyright and reuse:

This thesis is made available online and is protected by original copyright.

Please scroll down to view the document itself.

Please refer to the repository record for this item for information to help you to cite it.

Our policy information is available from the repository home page.

For more information, please contact the WRAP Team at: wrap@warwick.ac.uk



People Re-identification using Deep Appearance, Feature and Attribute Learning

by

Gregory Watson

Thesis

Submitted to the University of Warwick

for the degree of

Doctor of Philosophy in Urban Science

Department of Computer Science

January 2020

Contents

List of Tables	iv
List of Figures	vi
Acknowledgments	xiv
Declarations	xv
Abstract	xvi
Acronyms	xvii
Chapter 1 Introduction	1
1.1 Methods and Contributions	4
1.1.1 Partial Least Squares Appearance Modelling	4
1.1.2 Deep Foreground Appearance Modelling	4
1.1.3 Combining Deep Features and Attribute Detection for Re-ID	5
1.2 Thesis Outline	5
Chapter 2 Literature Review	6
2.1 Spatial & Foreground Modelling for Re-ID	6
2.2 Hand-Crafted Features	8
2.3 Deep Learning	10
2.3.1 A Background on Convolutional Neural Networks	10
2.3.2 Layers used commonly in Convolutional Neural Networks . .	14
2.3.3 Activation Functions	20
2.3.4 Training	21
2.3.5 Training Strategies	23
2.3.6 Transfer Learning	30
2.3.7 Evaluation	31
2.3.8 Cross-Validation	32

2.4	Attribute Learning	33
2.5	Other Related Works	36
2.5.1	Distance Metric Learning	36
2.5.2	Generative Adversarial Networks	37
2.6	Metrics	38
2.7	Data Sets	40
2.7.1	VIPeR (2007)	41
2.7.2	QMUL GRID (2009)	42
2.7.3	CUHK (2012-2014)	42
2.7.4	PRID2011 (2011)	43
2.7.5	3DPeS (2011)	43
2.7.6	i-LIDS (2009) and iLIDS-VID (2014)	44
2.7.7	Market-1501 (2015)	45
2.7.8	DukeMTMC-reID (2017) and DukeMTMC4reID (2017)	45
2.7.9	PETA (2014)	46
2.7.10	Limitations	47
2.8	Summary	48
Chapter 3 Partial Least Squares Appearance Modelling		49
3.1	Introduction	49
3.2	Partial Least Squares Foreground Appearance Modelling	50
3.3	Partial Least Squares Orientation Modelling	56
3.4	Feature Extraction and Weighting	59
3.4.1	LOMO	59
3.4.2	Salient Colour Names Based Colour Descriptor (SCNCD)	64
3.4.3	Weighted LOMO	66
3.5	Distance Metric Learning	68
3.6	Results and Discussion	69
3.6.1	Evaluation on the VIPeR data set	70
3.6.2	Evaluation on the QMUL GRID data set	77
3.6.3	Evaluation on the CUHK03 data set	83
3.7	Summary	85
Chapter 4 Deep Foreground Appearance Modelling		87
4.1	Introduction	87
4.2	Deep Neural-Network Appearance Modelling	88
4.3	Results and Discussion	91
4.3.1	Evaluation on the VIPeR data set	91
4.3.2	Evaluation on the QMUL GRID data set	101

4.3.3	Evaluation on the CUHK03 data set	110
4.4	Summary	112
Chapter 5 Combining Deep Features and Attribute Detection for Re-ID		115
5.1	Introduction	115
5.2	Deep Attribute Prediction	117
5.2.1	Deep Attribute Prediction Network	117
5.3	Results and Discussion	119
5.3.1	Training the Skeleton Prediction Model	120
5.3.2	Training the Attribute Prediction Model	120
5.3.3	Evaluation	122
5.3.4	Experimentation with different numbers of parts-based images	126
5.3.5	Class Imbalance	128
5.4	Summary	133
Chapter 6 Conclusions		135
6.1	Summary and Discussion	135
6.2	Future Work	138
Appendix A Data Annotation		142

List of Tables

3.1	The dimensionality of each of the feature type used in our experimentation. Due to the higher resolution available for most images within the CUHK03 [111] data set, we resize to 160×60 pixels for LOMO and Weighted LOMO feature extraction, rather than 128×48 pixels, resulting in a higher dimensional feature descriptor.	70
3.2	The VIPeR [59] data set was split into two sets, with 316 identities allocated for training and 316 for testing. Every probe image in the test set is compared to every gallery image in the test set. PLSAM(v2) consists of the Weighted LOMO and limb-by-limb level SCNCD feature descriptors with XQDA, whereas PLSAM(v1) consists just of the Weighted LOMO feature descriptors with XQDA.	76
3.3	The QMUL GRID [117, 124, 125] data set was split into two sets, with 125 identities allocated for training and 900 for testing. The 900 testing identities consisted of 125 image pairs and 775 single images. Every probe image in the test set is compared to every gallery image in the test set. PLSAM(v2) consists of the Weighted LOMO and limb-by-limb level SCNCD feature descriptors with XQDA, whereas PLSAM(v1) consists just of the Weighted LOMO feature descriptors with XQDA.	82
3.4	The CUHK03 [111] data set was split into two sets, with 1160 identities allocated for training and 100 for testing. Every probe image in the test set is compared to every gallery image in the test set. PLSAM(v2) consists of the Weighted LOMO and limb-by-limb level SCNCD feature descriptors with XQDA, whereas PLSAM(v1) consists just of the Weighted LOMO feature descriptors with XQDA.	84

4.1	Results on the VIPeR [59] data set. The best results are highlighted in bold. (v1) refers to the Weighted LOMO features and XQDA, whereas (v2) refers to the Weighted LOMO features with the limb-by-limb level SCNCD features and XQDA.	100
4.2	Results on the QMUL GRID [117, 124, 125] data set. The best results are highlighted in bold. (v1) refers to the Weighted LOMO features and XQDA, whereas (v2) refers to the Weighted LOMO features with the limb-by-limb level SCNCD features and XQDA.	109
4.3	Results on the CUHK03 [111] data set. The best results are shown in bold. (v1) refers to the Weighted LOMO features and XQDA, whereas (v2) refers to the Weighted LOMO features with the limb-by-limb level SCNCD features and XQDA.	111
5.1	Attribute detection accuracy on the VIPeR [59] data set. The ten best and worst attributes detection accuracies are shown.	120
5.2	Results on the VIPeR [59] data set. The best results are highlighted in bold. Results marked with * do not incorporate attribute features.	123
5.3	Results on the PRID2011 [70] data set. The best results are highlighted in bold. Results marked with * do not incorporate attribute features.	124
5.4	Results on the i-LIDS [57, 225, 226] data set. The best results are highlighted in bold. Results marked with * do not incorporate attribute features.	125
5.5	Results on the Market-1501 [219] data set. The best results are highlighted in bold. Results marked with * do not incorporate attribute features.	126
5.6	Results on the VIPeR [59] data set utilising different combinations of the original and parts-based images. Models are trained with BCE loss. The best results are highlighted in bold.	127
5.7	Results on the VIPeR [59] data set utilising WBCE loss and BCE loss. The best results are highlighted in bold.	130
5.8	Results on the PRID2011 [70] data set utilising WBCE loss and BCE loss. The best results are highlighted in bold.	130
5.9	Results on the i-LIDS [57, 225, 226] data set utilising WBCE loss and BCE loss. The best results are highlighted in bold.	131
5.10	Results on the Market-1501 [219] data set utilising WBCE loss and BCE loss. The best results are highlighted in bold.	131
5.11	The cosine distance between the attribute distribution of the training data set and each evaluation data set.	132

List of Figures

1.1	An example of the three stages of Re-ID. In stage 1, the person is localised from within the image frame. In stage 2, feature descriptors are extracted from each image. In stage 3, each input probe image is matched with the gallery image the smallest distance away in the feature space. Images with the same colour outline are of the same identity. Examples are taken from the PRW [223] data set.	2
1.2	Example images from Re-ID data sets [6–8, 57, 59, 70, 117, 125, 225, 226]. Each column represents a single identity. All images have been scaled to a standard resolution.	2
2.1	An example of the convolution operation applied on a 6×6 matrix with a 3×3 filter.	15
2.2	An example of the 3×3 filter in Figure 2.1 applied to an image. It can be observed that the filter highlights vertical edges within the image. In CNNs, the weights of the filter are learned by backpropagation. .	16
2.3	An example of max-pooling used to downsample a feature map. Within each pooling region, represented by a distinct colour, the maximum value is taken to form the corresponding value in the output feature map.	17
2.4	An example of average-pooling used to downsample a feature map. Within each pooling region, represented by a distinct colour, the average value of all values is taken to form the corresponding value in the output feature map.	17
2.5	A simple network containing only fully-connected layers. The network takes in input X_n of length 5, and predicts \hat{Y}_n of length 1.	18

2.6	A typical use-case for a CNN within the field of Re-ID. A pedestrian image from the VIPeR [59] data set is passed as input to a CNN, and is passed through a series of hidden layers to create a series of feature maps. Once the final feature map has been computed, it is passed to a fully-connected layer, where each unit within the fully-connected layer represents an identity. In this example, the network has predicted with 74% confidence that the identity of the person present in the input image is ID2.	19
2.7	Examples of (a) overfitting, (b) underfitting and (c) achieving a good fit on a data set.	23
	(a)	23
	(b)	23
	(c)	23
2.8	An example of a network trained using Verification Loss. The network is trained to predict whether or not a given pair of images represent the same identity or otherwise. The example images are taken from the VIPeR [59] data set.	25
2.9	An example of a network trained using ID Classification Loss. The network is trained to predict the identity of a given image. The example image is taken from the VIPeR [59] data set.	26
2.10	An example of a deep neural network utilising triplet loss. Three images are passed to the network, an <i>input image</i> , a <i>positive match</i> image with the same identity as the input image, and a <i>negative match</i> with a different identity to the input and positive match images. The network <i>pulls</i> the output of the input and positive match to be closer, by minimising the distance between the two within the output feature space. Furthermore, the network pushes the input and positive match away from the negative match, by maximising the distance between the different identities within the output feature space.	29
2.11	Example attributes and corresponding positive and negative examples. Images are taken from the VIPeR [59] data set with attribute labellings taken from the PETA [39] data set.	34

2.12	A comparison between rank- n and mAP/AP. It can be observed that whilst the rank-1 rate is consistently 100% throughout all examples, utilising mAP and AP better evaluates the performance when multiple images of the same identity as the input probe image are present within the gallery set. Images with the same identity as the input probe image are highlighted in green, whereas images with a different identity are highlighted in red. All images are from the Market-1501 [219] data set.	40
2.13	Example images from the VIPeR [59] data set. Each column represents a single identity. All images have been scaled to a standard resolution.	41
2.14	Example images from the QMUL GRID [117, 124, 125] data set. Each column represents a single identity. All images have been scaled to a standard resolution.	42
2.15	Example images from the CUHK01 [110] data set. Each column represents a single identity. All images have been scaled to a standard resolution.	43
2.16	Example images from the PRID2011 [70] data set. Each column represents a single identity. All images have been scaled to a standard resolution.	43
2.17	Example images from the 3DPeS [6–8] data set. Each column represents a single identity. All images have been scaled to a standard resolution.	44
2.18	Example images from the i-LIDS [57, 225, 226] data set. Each column represents a single identity. All images have been scaled to a standard resolution.	44
2.19	Example images from the Market-1501 [219] data set. Each column represents a single identity. All images have been scaled to a standard resolution.	45
2.20	Example images from the DukeMTMC-reID [152, 228] data set. Each column represents a single identity. All images have been scaled to a standard resolution.	46
2.21	Example images from the PETA [39] data set. All images have been scaled to a standard resolution. Individual images are originally part of [6–8, 12–14, 29, 57, 59, 70, 109–111, 117, 125, 142, 225, 226]. . . .	47
3.1	Examples of our PLS skeleton fitting on images from the VIPeR [59] data set. Each set of five images contains: the original image, the input HOG appearance features, the ground-truth skeleton keypoints, our predicted skeleton using the PLS regression model and the foreground image mask.	55

3.2	Examples of orientation groups from the VIPeR [59] data set. For this data set, we split the images into two orientation groups: those facing perpendicular (90°) to the camera, and those facing all other directions relative to the camera.	56
3.3	Examples of training the PLS appearance model on different subsets of the VIPeR [59] data set. The third column shows training only on images containing people with non-perpendicular orientations relative to the camera. The fourth column represents training on images containing only images containing people with perpendicular orientations relative to the camera. The fifth and final column shows training on images containing people with all poses relative to the camera. It can be seen that the lowest RMSE is obtained by training on images with similar orientations to the input image.	58
3.4	Example image pairs from the VIPeR [59] data set, containing both the original images as well as the images after the Retinex preprocessing step has been applied.	62
3.5	The LOMO [115] feature extraction method. The diagram is based on a similar diagram contained in [115].	64
3.6	SCNCD [208] feature extraction at a limb-by-limb level. The histograms below each individual limb image represent the extracted feature descriptor, with each bar representing an individual colour name.	66
3.7	Our proposed method for weighting the LOMO feature descriptor. Patch features are extracted from each 10×10 pixel region, and then weighted by multiplying by the percentage of foreground pixels. Then, we take all patches in the same horizontal location and use the maximum value of each bin to contribute towards the final feature descriptor for that horizontal location.	68
3.8	Examples of ground-truth and predicted skeletons from the VIPeR [59] data set. The average RMSE over the entire test set is 5.2 pixels. . .	71
3.9	The distribution of RMSE on skeletons predicted by the PLS-based skeleton prediction model on the VIPeR [59] data set.	72
3.10	Examples of ground-truth skeletons, and skeletons predicted using our PLS model on the VIPeR [59] data set which achieves a good skeleton fitting result.	73
3.11	Examples of ground-truth skeletons, and skeletons predicted using our PLS model on the VIPeR [59] data set which achieves a poor skeleton fitting result.	74

3.12	CMC on the VIPeR [59] data set. All of our CMC curves are single-shot results. Results are reproduced from [115], [217], [212], [111] and [113].	76
3.13	Examples of ground-truth and predicted skeleton from the QMUL GRID [117, 124, 125] data set. The average RMSE over the entire test set is 5.3 pixels.	78
3.14	The distribution of RMSE on skeletons predicted by the PLS-based skeleton prediction model on the QMUL GRID [117, 124, 125] data set.	79
3.15	Examples of ground-truth skeletons, and skeletons predicted using our PLS model on the QMUL GRID [117, 124, 125] data set which achieves a good skeleton fitting result.	80
3.16	Examples of ground-truth skeletons, and skeletons predicted using our PLS model on the QMUL GRID [117, 124, 125] data set which achieves a poor skeleton fitting result.	81
3.17	CMC on the QMUL GRID [117, 124, 125] data set. All of our CMC curves are single-shot results. Results are reproduced from [115], [217], [212], [111] and [113].	82
3.18	CMC on the CUHK03 [111] data set. All of our CMC curves are single-shot results. Results are reproduced from [115], [217], [212], [111] and [113].	84
4.1	The network architecture for our proposed deep foreground modelling network. We first rescale images to a resolution of 224×224 pixels, and pass the images through the convolutional layers of a ResNet-50 network [66]. We take the POOL5 average pooling layer as the output of the ResNet-50 model, flatten the output, and pass the output through two further fully-connected layers. The output of our proposed network contains 58 units, representing the (x, y) coordinates of the skeleton key-points (joints and edge markers). We use the RMSProp [177] optimizer, with a mean squared error loss.	89
4.2	Examples of images, skeletons, and their corresponding augmentations. The first image in each row is the original image. The remaining images in each row are augmentations of the first image.	90
4.3	Examples of ground-truth, skeletons predicted using our deep model, and skeletons predicted using our PLS model on the VIPeR [59] data set. The average RMSE when using the deep model was 4.5 pixels, whilst the average when using the PLS model was 5.2 pixels.	93

4.4	The distribution of RMSE on skeletons predicted by the deep skeleton prediction model on the VIPeR [59] data set. The average RMSE was 4.5 pixels, whilst the average using the PLS model was 5.2 pixels. . .	94
4.5	Examples of ground-truth, skeletons predicted using our deep model, and skeletons predicted using our PLS model on the VIPeR [59] data set which achieve a good skeleton fitting result.	95
4.6	Examples of ground-truth, skeletons predicted using our deep model, and skeletons predicted using our PLS model on the VIPeR [59] data set which achieve a poor skeleton fitting result.	96
4.7	A comparison of skeleton fitting results using our deep method against the fittings labelled as poor in Figure 3.11.	97
4.8	Further comparisons of skeleton fitting results using our deep method against the fittings labelled as poor in Figure 3.11.	98
4.9	CMC on the VIPeR [59] data set. All of our CMC curves are single-shot results. Results are reproduced from [77, 111, 113, 115, 189, 212, 217].	101
4.10	Examples of ground-truth, skeletons predicted using our deep model, and skeletons predicted using our PLS model on the QMUL GRID [117, 124, 125] data set. The average RMSE when using the deep model was 5.5 pixels, whilst the average when using the PLS model was 5.3 pixels.	103
4.11	The distribution of RMSE on skeletons predicted by the deep skeleton prediction model on the QMUL GRID [117, 124, 125] data set. The average RMSE was 5.5 pixels, whilst the average using the PLS model was 5.3 pixels.	104
4.12	Examples of ground-truth, skeletons predicted using our deep model, and skeletons predicted using our PLS model on the QMUL GRID [117, 124, 125] data set which achieve a good skeleton fitting result using our deep model.	105
4.13	Examples of ground-truth, skeletons predicted using our deep model, and skeletons predicted using our PLS model on the QMUL GRID [117, 124, 125] data set which achieve a poor skeleton fitting result using our deep model.	106
4.14	A comparison of skeleton fitting results using our deep method against the fittings labelled as poor in Figure 3.16.	107
4.15	Further comparisons of skeleton fitting results using our deep method against the fittings labelled as poor in Figure 3.16.	108

4.16	CMC on the QMUL GRID [117, 124, 125] data set. All of our CMC curves are single-shot results. Results are reproduced from [77, 111, 113, 115, 189, 212, 217].	109
4.17	CMC on the CUHK03 [111] data set. All of our CMC curves are single-shot results. Results are reproduced from [77, 111, 113, 115, 189, 212, 217].	112
5.1	An example of images from the VIPeR [59] data set. All of these images are labelled as wearing blue clothing on their upper bodies by the PETA [39] data set. However, significant visual variation can be seen between images, such as pose and illumination variation.	116
5.2	An example of how we divide each Re-ID image into three parts-based images: top, middle and bottom, using our deep CNN-based method proposed in Chapter 4. We create a bounding box around each part, and add padding of 15% in the x and y dimensions, to account for any errors in skeleton prediction. We use the original image and the three parts-based images as input to our attribute prediction model. (a) The original input image; (b) The original image with the skeleton and parts separation overlaid; (c) The individual body parts images.	118
5.3	The network architecture of the attribute prediction model. The original image is divided into three body parts - the top, middle and bottom. The original image, as well as the three body parts images, are passed through an identical ResNet-50 [66] network architecture. The fully-connected layers of each ResNet-50 model are removed and replaced with our own fully-connected layer of size 512. The four fully-connected layers are then concatenated to form a layer of size 2048. Finally, we append a fully-connected layer of size n , with n being the number of attributes being predicted.	119
5.4	Examples of attribute prediction accuracy. All images shown in (a) are predicted to be wearing red clothing on their upper body, whilst images in (b) are predicted to be wearing a backpack. Images correctly classified (true-positive) are highlighted in green, whilst those incorrectly classified (false-positive) are highlighted in red. The predicted probability of the presence of each attribute is shown below each image.	121
5.5	The distribution of attributes on the data sets used to train the attribute model, versus the three data sets used to evaluate the attribute model.	132

A.1	Example images and their corresponding hand-labelled skeletons from the VIPeR [59] data set.	143
A.2	An example of the skeleton labelling GUI using images from the VIPeR [59] data set. The first Re-ID image represents the input image on which the user clicks to mark skeleton keypoints. The second Re-ID image shows the recorded skeleton keypoints converted to a skeleton and overlaid on the input image in real-time. The third image is a static reference image showing the order in which the skeleton keypoint should be collected. Finally, some information on the Re-ID image is shown on the right of the GUI.	144

Acknowledgments

Firstly, I would like to express my sincere gratitude to my supervisor, Dr. Abhir Bhalerao, for the continuous support that he has provided throughout my study. I have benefited greatly from his encouragement and advice, as well as with his immense knowledge and experience. I would also like to thank Dr. Matthew Leeke and Prof. Nasir Rajpoot for acting as advisors throughout the project.

Furthermore, I would like to thank my friends and colleagues in the Warwick Institute for the Science of Cities and Department of Computer Science, Katherine Ascott, Jamie Bayne, Matthew Bradbury, James Van Hinsbergh, Melissa Kenny, Richard Kirk, Helen McKay, David Purser, John Rahilly, Liam Steadman, David Truby and Ian Tu for the support, laughs and great times that we shared. I would additionally like to thank the support staff, Yvonne Colmer, Katie Martin and Jenny Eborall, who have helped significantly over the past years with support and organisational matters.

Lastly, I would like to thank my family and Andrew Brooks, for their support and encouragement throughout my PhD study.

Declarations

I declare that the work presented in this thesis, titled *Person Re-identification using Deep Appearance, Feature and Attribute Learning*, is original work, and has not been submitted for a degree at another university. Parts of this thesis have been previously published in the following:

- Gregory Watson and Abhir Bhalerao. Person re-identification using partial least squares appearance modelling. In *International Conference on Image Analysis and Processing*, pages 25–36. Springer, 2017. doi: 10.1007/978-3-319-68548-9_3
- Gregory Watson and Abhir Bhalerao. Person reidentification using deep foreground appearance modeling. *Journal of Electronic Imaging*, 27(5):051215, 2018. doi: 10.1117/1.JEI.27.5.051215
- Gregory Watson and Abhir Bhalerao. Person re-identification combining deep features and attribute detection. *Multimedia Tools and Applications*, December 2019. doi: 10.1007/s11042-019-08499-9

Abstract

Person Re-Identification (Re-ID) is the act of matching one or more query images of an individual with images of the same individual in a gallery set. We propose various methods to improve Re-ID performance via foreground modelling, skeleton prediction and attribute detection.

Foreground modelling is an important preprocessing step in Re-ID, allowing more representative features to be extracted. We propose two foreground modelling methods which learn a mapping between a set of training images and skeleton keypoints. The first utilises Partial Least Squares (PLS) regression to learn a mapping between Histogram of Oriented Gradients (HOG) features extracted from person images, and skeleton keypoints. The second instead learns the mapping using a deep convolutional neural network (CNN). Using a CNN has been shown to generalise better, particularly for unusual pedestrian poses.

We then utilise the predicted skeleton to generate a binary mask, separating the foreground from the background. This is useful for weighting image features extracted from foreground areas higher than those extracted from background areas. We apply this weighting during the feature extraction stage to increase matching rates.

The predicted skeleton can be used to divide a pedestrian image into multiple parts, such as head and torso. We propose using the divided images as input to an attribute prediction network. We then use this network to generate robust feature descriptors, and demonstrate competitive Re-ID matching rates.

We evaluate on a number of different Re-ID data sets, each possessing significant variations in visual characteristics. We validate our proposals by measuring the rank- n score, which is equivalent to the percentage of identities correctly predicted within n attempts. We evaluate our skeleton prediction network using root mean square error (RMSE), and our attribute prediction network using accuracy. Experiments demonstrate that our proposed methods can supplement traditional Re-ID approaches to increase rank- n matching rates.

Acronyms

- AP** Average Precision.
- AUC** Area Under The Curve.
- BCE** Binary Cross-Entropy.
- CAN** Comparative Attention Network.
- CCA** Canonical Correlation Analysis.
- CCE** Categorical Cross-Entropy.
- CMC** Cumulative Matching Characteristic.
- CNN** Convolutional Neural Network.
- CPS** Custom Pictorial Structures.
- CRF** Colour Restoration Function.
- DFAD** Deep Features & Attribute Detection.
- DGD** Domain Guided Dropout.
- DNAM** Deep Network Appearance Model.
- DPM** Deformable Part Model.
- FAN** Feature Aggregation Network.
- FN** False Negative.
- FP** False Positive.
- FPNN** Filter Pairing Neural Network.
- GAN** Generative Adversarial Network.
- GOG** Gaussian of Gaussian.

GUI Graphical User Interface.

HOG Histogram of Oriented Gradients.

IBP Indian Buffet Process.

JLML Jointly Learning Multi-Loss.

LNCC Local Normalized Cross-Correlation.

LOMO Local Maximal Occurrence.

LRN Local Response Normalization.

LSR Label Smoothing Regularization.

LSRO Label Smoothing Regularization for Outliers.

LSTM Long Short-Term Memory.

MAE Mean Absolute Error.

mAP Mean Average Precision.

MLP Multilayer Perceptron.

MLR Multiple Linear Regression.

MSCR Maximally Stable Color Regions.

MSE Mean Squared Error.

MTL Multi-Task Learning.

OLS Ordinary Least Squares.

PCA Principal Component Analysis.

PCR Principal Component Regression.

PLS Partial Least Squares.

PLSAM Partial Least Squares Appearance Model.

PRDC Probabilistic Relative Distance Comparison.

PS Pictorial Structures.

RCN Recurrent Comparative Network.

Re-ID Person Re-Identification.

RHSP Recurrent High-Structured Patches.

RMSE Root Mean Squared Error.

RNN Recurrent Neural Network.

ROC Receiver Operating Characteristic.

SCA Stel Component Analysis.

SCNCD Salient Color Names Based Color Descriptor.

SIFT Scale-Invariant Feature Transform.

SILTP Scale Invariant Local Ternary Pattern.

SURF Speeded-Up Robust Features.

SVD Singular Value Decomposition.

SVM Support Vector Machine.

TN True Negative.

TP True Positive.

WBCE Weighted Binary Cross Entropy.

XQDA Cross-view Quadratic Discriminant Analysis.

Chapter 1

Introduction

Person Re-Identification (Re-ID) is the process of automatically identifying and matching different images of people taken from separate, non-overlapping cameras at different times. It has a multitude of important applications in security, surveillance and biometrics, as well as in tracking and people-monitoring.

Re-ID can be broken down into three stages. The first step is to localise the person within the image of the environment, typically by using either a pedestrian detector [111, 219, 220, 224] or via hand-labelling [59, 111, 117, 125, 219]. The second part concerns extracting a robust feature descriptor of the person. A lack of high-resolution imagery in Re-ID data sets have rendered more traditional biometric approaches such as facial recognition unsuitable, leaving low level features either pre-specified by the user, such as colour [43, 58, 115, 117, 130, 148, 208, 216] or texture [43, 58, 115, 117], or those learnt via a deep convolutional neural network (CNN) [2, 27, 28, 51, 90, 108, 111, 112, 200, 203, 206, 229, 230]. The third and final part relates to matching [115, 156, 185, 206, 212], where the distance between the extracted feature descriptors is computed. Whilst some techniques calculate the Euclidean distance between these feature descriptors, this distance metric is often learnt through supervised learning. Given two feature descriptors extracted from different images of the same person, it is expected that the distance between the two is smaller than the distance between two features descriptors extracted from different people. An illustrative example of the three stages of Re-ID can be seen in Figure 1.1.

However, Re-ID is often challenging due to the variations in pose, illumination and image resolution caused by the use of distinct, non-overlapping cameras [24, 53, 188]. The extracted feature descriptors are often not invariant to the significant inter-class and intra-class variations often present in Re-ID images. Figure 1.2 shows examples of images from various common Re-ID data sets, and demonstrates the

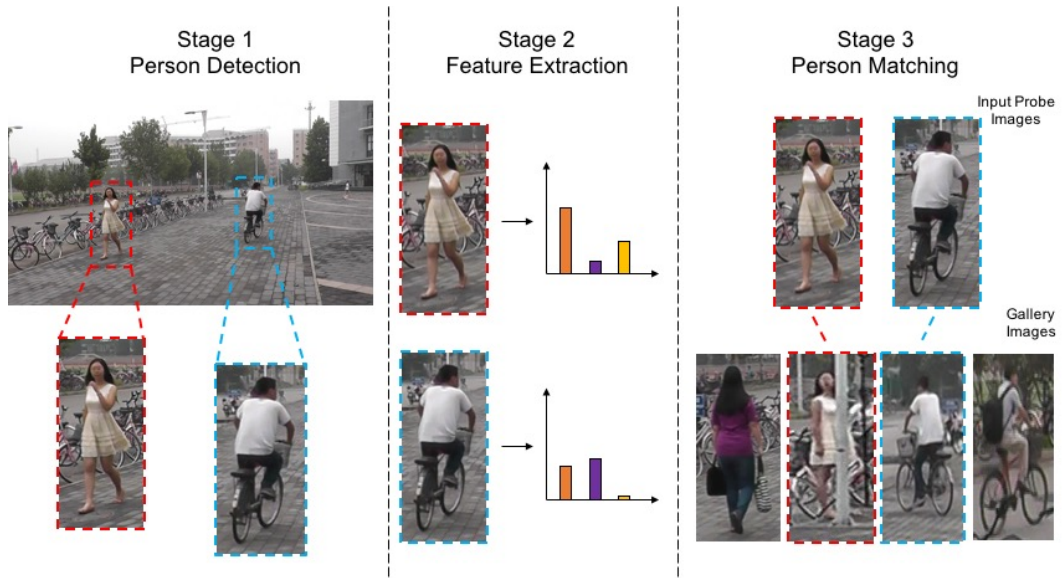


Figure 1.1: An example of the three stages of Re-ID. In stage 1, the person is localised from within the image frame. In stage 2, feature descriptors are extracted from each image. In stage 3, each input probe image is matched with the gallery image the smallest distance away in the feature space. Images with the same colour outline are of the same identity. Examples are taken from the PRW [223] data set.

types of variation found in these images.

One of the ways in which the negative effects of variations in visual characteristics can be minimised is to pose normalise the images. A significant variation between images is the pose of the person, with some people having been photographed from the front, whilst others have been photographed from the side. Furthermore, across images, limbs such as arms and legs can be located in a wide variety of positions relative to the torso. The variance in the position of a persons body is directly linked to the variation in the position of the background, largely redundant information which is not representative of the person.

One of the ways in which pose and background variation can be minimised is to use a foreground appearance model, built using a model-based approach such as Partial Least Squares (PLS) [127, 155] regression. Utilising this technique, a regression can be learnt between image appearance features, as well as landmarks representing foreground regions of a Re-ID image.

Inspired by the processes of the human brain, an alternative approach is to use methods to *learn* the optimal mapping between image appearance and foreground landmarks. A specific branch of machine learning, known as deep learning [55, 95, 160], achieves this by using an artificial neural network, which consist of a series of hierarchical layers, including input, output, and “hidden” layers



Figure 1.2: Example images from Re-ID data sets [6–8, 57, 59, 70, 117, 125, 225, 226]. Each column represents a single identity. All images have been scaled to a standard resolution.

performing operations such as convolution and pooling. Each layer is connected to the next with a series of weights. During training, the network predicts the corresponding output for each input image, and compares its prediction to the ground-truth labels. The network then adjusts its layer weights based on its prediction performance relative to the ground-truth labelling, with the overall goal being to reduce the loss.

However, significantly more training data is required to train a deep neural network compared to more traditional, less complex methods such as PLS. Yet, Re-ID data sets are small in size, with very few images per identity. Furthermore, Re-ID data sets with foreground labelling are uncommon, with most data sets simply consisting of an image and identity labelling. Therefore, training a deep neural network only on Re-ID data sets is often not the best way to achieve a high-performing network. It is commonplace for networks to instead be trained on a problem from a different domain, such as object recognition, with transfer learning [147] then used to apply the learnt knowledge to the problem of Re-ID. For example, the ResNet-50 [66] network trained on the ImageNet [38] data set is often used as a basis for networks involved in the field of Re-ID.

In this thesis, we focus on the task of minimising the effects of pose and background variations through the use of person appearance modelling, which is a form of foreground modelling. By learning the foreground, we are able to predict the skeleton of an individual when presented with their image. This information can then be used to address specifically the issue of pose variation by providing context to our feature extraction process. We begin with skeleton prediction through

utilising a PLS-based [127, 155] model. We then employ end-to-end learning to train a CNN to perform the skeleton prediction task. Finally, we combine our skeleton prediction models with attribute information to improve feature engineering, creating features which are more invariant to variations in visual characteristics. We use these techniques to supplement and improve existing Re-ID techniques to increase matching rates, and validate our work on standard Re-ID data sets.

1.1 Methods and Contributions

We make three contributes within this thesis. Two contributions relate to improving feature extraction by assimilating foreground information learnt via a skeleton prediction model, whilst the third relates to improving Re-ID matching rates by incorporating feature descriptors learnt through an attribute prediction method.

1.1.1 Partial Least Squares Appearance Modelling

We propose a novel skeleton prediction method using PLS regression, which learns a mapping between image appearance information and skeleton keypoints. By predicting skeleton information for a given image, we can exploit this information during the feature extraction stage to prioritise feature descriptors extracted from foreground regions. We alter Local Maximal Occurrence (LOMO) [115] features to create Weighted LOMO, which weights LOMO feature descriptors extracted from foreground areas higher than those extracted from background areas. We concatenate with Salient Colour Names Based Colour Descriptor (SCNCD) [208] feature descriptors extracted on a limb-by-limb level, and follow by utilising the Cross-view Quadratic Discriminant Analysis (XQDA) [115] Distance Metric Learning technique to calculate the distance between feature descriptors.

1.1.2 Deep Foreground Appearance Modelling

Following on from the previous work using PLS to learn a mapping between image appearance information and skeleton keypoints, we propose the replacement of the PLS method with a deep CNN. Whilst the PLS approach struggles to handle person images taken from different angles, and thus requires multiple models to be trained, the Deep CNN approach is able to predict the skeleton of person images with different poses with a greater degree of accuracy, whilst only requiring a single skeleton prediction model to be trained. Furthermore, we find that our deep CNN-based method is able to better handle the individuals with more unusual poses, such as those with arms raised. We validate the predicted skeletons in a Re-ID framework, and show improvement in matching rates.

1.1.3 Combining Deep Features and Attribute Detection for Re-ID

We present a CNN-based architecture which takes as input a person image, as well as three parts-based images generated via our Deep Network Appearance Model, and predicts a set of fifty attributes. We then use the penultimate layer in the network as a feature descriptor for matching. Notably, none of the data sets which we use to evaluate our method contribute training images to the attribute network, demonstrating our method’s ability to generalise between different data sets. Furthermore, given that some attributes will be more prevalent than others within the training set, we propose a novel Weighted Binary Cross Entropy function that weights the cost of a positive error relative to a negative error, based on the ratio of positive to negative instances of each attribute. We evaluate the combined approach on a set of standard data sets.

1.2 Thesis Outline

In Chapter 2, we perform a review of the literature related to Re-ID. We review approaches which utilise spatial & foreground modelling, hand-crafted features, as well as attribute learning. We also detail metrics used to evaluate Re-ID methods, as well as the data sets on which they are evaluated. In Chapter 3, we describe our method which utilises PLS regression to learn a mapping between image appearance information and skeleton keypoints, allowing feature descriptors to be extracted primarily from foreground regions prior to matching. In Chapter 4, we replace our PLS-based skeleton prediction model with a deep CNN-based approach, and compare both the PLS and CNN-based methods. In Chapter 5, we propose utilising skeleton information as context to an attribute prediction network, which is then used as a feature extraction network for matching. Finally, in Chapter 6, we summarise our proposed methods and describe potential future research directions.

Chapter 2

Literature Review

In this chapter, we discuss and review previous Re-ID techniques, paying particular attention to spatial modelling, feature extraction, deep learning, attribute detection, distance metric learning, and the use of generative adversarial networks.

2.1 Spatial & Foreground Modelling for Re-ID

High accuracy Re-ID results depend on the ability to produce robust feature descriptors which possess low intra-class variation and high inter-class variation, and so, the feature extraction stage is of vital importance. However, extracting feature descriptors from images without attention given to significant background differences can lead to the extracted feature descriptors being affected by extraneous and unrepresentative background information. Much study [29, 43, 108, 135, 136, 172, 199, 215, 222] has therefore been carried out to investigate methods for separating the foreground from the background, allowing feature descriptors to be extracted either in full or primarily from foreground areas rather than background areas.

Stel Component Analysis [86] (SCA) has been utilised to separate the foreground of person images from the background. SCA attempts to capture the structure of an image by separating the image into a series of parts, or *stels*, that possess a similar feature distribution, such as colour or texture. However, if a single part of an image possesses a wide feature distribution, or if two parts of an image possess a similar feature distribution, separation into distinct parts may fail. Farenzena et al. [43] incorporate SCA and divide each person image into three distinct parts according to the person's limbs, calculating the vertical central axis for each part, and weighting each pixel dependent on the distance to the vertical central axis via a Gaussian kernel. This generally works well given that the pixels closer to a person's vertical central axis are much more likely to belong to their body, however this may fail if a person has their legs spread widely apart.

Spatial and foreground modelling can also help counter the problems of pose variation. Extracting feature descriptors without paying due attention to pose variations between images can lead to lack of spatial correspondence between the extracted feature descriptors. In Local Maximal Occurrence (LOMO) [115], the authors extract a histogram-based feature descriptor from 10×10 pixel patches, with an overlap of 5 pixels in each dimension. For each row of feature descriptors extracted, they take the maximum value in each histogram bin to form the final feature descriptor for that row. Whilst not explicitly removing or deprioritising the background information like alternative methods, the authors argue that their method achieves some invariance to viewpoint and pose changes by maximising the local occurrence of each histogram bin.

Other methods have attempted to predict the foreground of a person image via limb or skeleton prediction. For example, Cheng et al. [29] use an off-the-shelf Pictorial Structures (PS) [4] method to predict the location of six person limbs (head, chest, left and right thigh, and left and right leg). The authors then proposed an extension to PS which takes advantage of multiple images of an individual to improve the limb-fitting, result in Custom Pictorial Structures (CPS). Su et al. [172] use a human pose estimation algorithm to estimate the skeleton, and then from this they estimate the position of human body parts. These body part regions are then transformed to create a modified parts image with the body parts placed in standardised location between images. In order to roughly maintain the original aspect ratio of each body part, the modified parts images contain significant unavoidable blank space around each part.

Skeleton information is also often used in collaboration with depth information. Munaro et al. [135] use RGB-D images combined with skeleton information obtained from a Microsoft Kinect camera. Using this skeleton information, the authors transform the RGB-D images to a standard pose, building a standardised 3D pointclouds for each person. The standardised 3D pointclouds are then compared as part of the matching process. The authors also propose the use of multiple RGB-D images of the same person to build a single standardised 3D pointcloud, countering the potential effects of illumination variation and occlusion. Wu et al. [199] extended this idea but instead propose a method which uses RGB person images combined with skeleton information to predict the depth of the person image, allowing a pointcloud to be used for matching. The combination of RGB appearance features with learned depth features is demonstrated to increase matching rates. Munaro et al. [136] use two skeleton tracking algorithms, the Kinect Skeletal Tracker (KST) [166], and the Nite Skeletal Tracker (NST) [139, 140], which detect 20 and 15 points on the persons body respectively. The NST also takes advantage of multiple frames to improve the

tracking performance. Texture, colour and three-dimensional space features typically used in the field of robotics are then extracted from around the predicted skeleton points, and concatenated to build the final feature descriptor, leading to increased matching rates. Zheng et al. [222] predict a set of fourteen body joints, such as head and neck, using a Convolutional Pose Machine (CPM) [193]. These fourteen body joints are then used to localise ten body parts, which are projected to rectangles using affine transformations. The authors then combine these projected rectangles to form a PoseBox, which consists entirely of body parts with no blank space. Zhao et al. [215] instead learn a set of part-maps across a training set of images, with each part map relating to an individual body part across all images. Features are then extracted from the areas represented by the part maps. This method allows for body parts to be shaped in ways other than rectangles, and can detect the same body part between images even when these body parts are in different physical locations.

Other methods crop groups of pedestrian limbs to their bounding boxes, rather than to remove the background entirely. Li et al. [108] propose a Latent Part Localization model, removing significant amounts of background when the image is not well cropped to the person. A Spatial Transformer Network (STN) [83] is used to learn an appropriate crop as part of the training stage, which is then used to crop the person image into three parts, roughly corresponding to the head, torso and legs respectively. Each of these parts-based images are then passed through separate CNNs, learning optimal features for each part and concatenating to produce the final feature descriptor.

2.2 Hand-Crafted Features

Re-ID results are highly dependent on the extraction of robust feature descriptors which possess low intra-class variation and high inter-class variation. Thus, various different feature types have been proposed for the task of Re-ID. These proposed feature types must be invariant to the common issues faced in the problem of Re-ID, including but not limited to pose and illumination variation.

Hand-crafted features are defined as feature descriptors where the type of information extracted has been manually specified, and consists of an explicitly measurable quality of an image, such as colour or texture. This is in contrast to deep features (Section 2.3), where the most optimal type of information is learned by a neural network. Given that the type of feature extracted is pre-defined, unlike deep features, hand-crafted feature extraction does not require a training set for learning an appropriate feature type.

The most commonly used hand-crafted feature type is colour [43, 58, 115,

117, 130, 148, 208, 216]. Colour provides a simple and easily measurable means of extracting feature descriptors, and is generally invariant to non-significant variations in pose, illumination and background. Also, texture information [43, 58, 115, 117] is also commonly extracted from Re-ID images, being particularly discriminative for individuals who are wearing patterned clothing. For example, in [58], colour histograms in the RGB, YCbCr and HSV colour spaces, as well as texture histograms using Gabor [45] and Schmid [159] texture filters are utilised for building feature descriptors. Furthermore, Farenzena et al. [43] propose the use of Weighted Colour Histograms, where pixels closer to the vertical central axis of a person’s body count more towards the final colour histogram, the expectation being that the final histogram will be more representative of the person rather than the background region. The authors additionally extract Maximally Stable Color Regions (MSCR) features [46], which divides an image into a set of blob regions, with each blob region containing pixels of a similar colour. Each blob region is then described by its area, centroid, second moment matrix and average colour. The authors also propose their own feature descriptor, named Recurrent High-Structured Patches (RHSP). This feature type highlights image patches within the person image that reoccur. Patches are selected based on the amount of structural information present within them, and as such, a patch with a low amount of structural information, such as a patch which contains only solid colour, is discarded. The authors achieve this by calculating the sum of the entropy of each of the RGB channels for each patch, and prune out patches with a value lower than a certain threshold. The Local Normalized Cross-Correlation (LNCC) is then calculated for each patch, and patches with high LNCC values are clustered together. For each cluster, the patch closest to the cluster’s centroid is considered a recurrent patch. Matsukawa et al. [130] propose the Gaussian of Gaussian (GOG) descriptor, which captures both mean and covariance information of pixels within image patches. The authors achieve this by modelling each set of patches as a series of multiple Gaussian distributions, where each Gaussian distribution represents an individual patch. This overall set of Gaussians is then used as a feature descriptor. To capture the information present within each patch, the authors extract the patches location, textural gradient information, as well as the RGB colour channels.

Interest points-based features have also been utilised in Re-ID. In [88], an Implicit Shape Model (ISM) [107] is built which uses Scale-Invariant Feature Transform (SIFT) [123] features for tracking. SIFT features are passed to a clustering algorithm, which identifies re-occurring features within the images. The cluster centres then act as prototypes for a codebook. SIFT features extracted from input images are matched with these prototypes. In [216], 32-bin colour histograms from the

LAB colour space, as well as SIFT features, are extracted from a set of 10×10 pixel patches with an overlap of four pixels. In addition, the colour histograms are extracted with three levels of downsampling with scaling factors of 0.5, 0.75 and 1. Their final feature vector, named dColorSIFT, consists of the concatenated colour histograms and SIFT features. Hamdoun et al. [63] instead extract Speeded-Up Robust Features (SURF) [9], an interest point which uses an integral image to increase the efficiency of interest point computation compared to SIFT.

Liao et al. [115] counteract the issue of illumination variation by utilising the Retinex transform [84, 85, 98], which typically results in much more vivid images with clearer detail, particularly in areas of shadow. This can result in a more standardised set of images even between multiple cameras with significant illumination variation. The authors then extract both HSV colour histograms and Scale Invariant Local Ternary Pattern (SILTP) [114] texture histograms. These features are calculated for each 10×10 pixel patch, with an overlap of 5 pixels in each dimension. To address the issue of pose variation, the authors analyse each row of patches, and take the highest value in each histogram bin to form the final histogram descriptor for that row. The histogram descriptors for each row are then concatenated to create a feature descriptor for the entire image. Afterwards, the image is then downsampled by a factor of two and four, and the process is repeated. The feature descriptors for both the original and downsampled images are then concatenated to form the final feature descriptor.

2.3 Deep Learning

More recent work has typically shifted from hand-crafted features to utilising deep neural networks [2, 27, 28, 51, 108, 111, 112, 200, 203, 206, 229, 230]. Hand-crafted features can sometimes be suboptimal for a given problem, especially when significant intra-class variations are present. For example, colour can appear significantly different on a day with bright sunlight compared with at night. Whilst hand-crafted features consist of a manually specified, measurable characteristic of an image, the parameters of deep networks are instead learnt to be optimal for a given problem.

2.3.1 A Background on Convolutional Neural Networks

McCulloch and Pitts [131] introduced the first model of an artificial neuron, proposing that neural events and the relations between neurons can be represented by propositional logic. Their work, whilst primitive compared to artificial neural networks of today, is often considered to be the building blocks of modern neural networks. However, given a set of input variables, the network proposed by McCul-

loch and Pitts [131] considers each input to have equal importance. Consider the task of predicting whether or not a store will sell a hundred ice creams on a given day, with the input consisting of the *weather*, *the day of the week* and *the proximity to the sea*. It may be the case that a store close to the sea may be visited by a large number of holidaymakers, and hence may sell a large number of ice creams regardless of weather or day of the week. However, a store further inland may sell significantly more ice creams on warmer days than otherwise. Hence, each input variable should be weighted, with the weights being dependent on the data. Rosenblatt [153, 154] built upon the work of McCulloch and Pitts [131] by proposing the Perceptron. The Perceptron introduces the concept of variable weights to the network, allowing different input variables to be weighted according to their importance. In addition, unlike the work introduced by McCulloch and Pitts [131], the Perceptron is able to *learn* the most appropriate weights. However, both methods are only able to classify linear data.

A convolutional neural network (CNN) [105] is a multi-layer feed-forward neural network typically used for computer vision classification tasks, based on the visual cortex of mammals. They have become widely used in recent years for learning optimal deep features for Re-ID [27, 28, 108, 203, 206, 229]. The experiments of Hubel and Wiesel [78, 79, 80], which inserted a microelectrode into the primary visual cortex of an anesthetized cat, are widely considered to have laid the groundwork for CNNs. By projecting various different sizes and shapes of light onto a screen, the authors found that some neurons in the cats brain only responded to the light at certain orientations presented within the neurons' receptive field, whilst other neurons responded only to alternate orientations. Furthermore, these types of cells demonstrated limited spatial invariance, and could be divided into excitatory and inhibitory regions. Hubel and Wiesel named these types of cells *simple cells*. They also identified a further type of cells present *complex cells*, which contain a larger receptive field and display a higher level of spatial invariance, and could not be divided into excitatory and inhibitory regions.

Fukushima [49] expanded upon the work of Hubel and Wiesel by introducing the Neocognitron, a multi-layer artificial neural network focused on the task of handwritten character recognition, as well as similar pattern recognition tasks. Their work introduced both convolutional and pooling layers for use in neural networks. LeCun et al. [106] proposed the use of backpropagation as a gradient-based learning technique for use with CNNs, also using their network for handwritten character recognition, to great success.

Whilst CNNs were therefore noted for their high performance in computer vision tasks in 1990s, they were not widely used because of the lack of computational

power. In the early 2010s, computational power had increased significantly, and the first modern convolutional neural network was released by Krizhevsky et al. [95], named *AlexNet*. Recent artificial neural networks typically consist of a large number of layers (Section 2.3.2), such as convolutional layers, pooling layers and fully-connected layers. The large number of layers make these networks *deep*, and hence this approach is referred to as *deep learning* [55].

Formally, a neural network utilising *supervised learning* takes an input X , and target Y , and learns a function F , where F is a function mapping X to Y : $\hat{Y} = F(X)$. This is in contrast to *unsupervised learning*, which takes an input X , but does not possess targets Y , and instead determines patterns within the input data, constructing a series of classes based on this information [17, 103, 150].

The layers of the network implement their own functions $X_l = F_l(X_{l-1})$, where X_l is the output of layer l for $l \in (0, 1, \dots, L - 1)$ for a network with L layers. Data sets are divided into training, validation, and testing sets. The data within the training set are used to train, and hence optimise the values of the weights, of a network. Whilst the data from the validation set is not used to optimise the values of the weights, it is used to ensure the generalisation ability of the network and prevent overfitting during the training stage, by evaluating the performance of the network on a set of data different from that used to train the network. Given a neural network and a set of ground-truth input-output pairs (X_n, Y_n) for $n \in (0, 1, \dots, p - 1)$, where p is the number of ground-truth input-output pairs in the training set, the goal of the network is to learn a mapping between X and Y which minimises the value of a loss function ψ , a function which evaluates the performance of the network during training. Given a list of input variables X , corresponding ground-truth labels Y , we can compute the error of the network, ϵ , by:

$$\epsilon = \psi(Y, \hat{Y}) \tag{2.1}$$

where \hat{Y} represents the output as predicted by the network. The choice of loss function, ψ , is highly dependent on the type of data being predicted by the network. For example, if the network is tasked with predicting a continuous value, such as the price of a house based on a number of different input factors, mean squared error (MSE) could be used. Given a ground-truth vector Y and predicted vector \hat{Y} , both of length N , MSE is defined as:

$$\epsilon_{MSE}(Y, \hat{Y}) = \frac{1}{N} \sum_{i=0}^{N-1} (Y_i - \hat{Y}_i)^2 \tag{2.2}$$

MSE is useful for computing the error between two vectors, and is therefore suited to pose and skeleton prediction [10, 97]. However, consider a network which

predicts the class of an input from a set of possible classes. In this case, the final layer of will consist of a fully-connected layer with a softmax activation function. As shown in Figure 2.6 each unit in the layer represents a specific class/identity, with the probabilities allocated to each class adding up to one. The softmax activation function, $\sigma_{softmax}$, can be defined as:

$$\sigma_{softmax}(x)_d = \frac{e^{x_d}}{\sum_{c=1}^C e^{x_c}} \quad (2.3)$$

where $d \in (1, 2, \dots, C)$, and C is the number of classes. As the task is a classification problem, it has C categorical outputs, rather than outputs over a continuous range. For this purpose, a loss function such as Categorical Cross-Entropy (CCE) is more appropriate. Within Re-ID, this loss function is often used for ID classification (Section 2.3.5), which involves predicting an identity from a typically large number of identities [108, 112, 203, 206]. The CCE between two arrays Y and \hat{Y} is defined as:

$$\epsilon_{CCE}(Y, \hat{Y}) = -\frac{1}{N} \frac{1}{C} \sum_{i=1}^N \sum_{j=1}^C Y_{ij} \log(\hat{Y}_{ij}) \quad (2.4)$$

where C is the number of classes, Y is an array of length N consisting of one-shot ground-truth vectors with C possible classes, and \hat{Y} is an array of length N consisting of softmax probability distributions output by the network representing the probability that the input vector belongs to each of the C classes. Similarly, Y_{ij} represents the i th vector within the array and the j th class within the vector. \hat{Y}_{ij} represents the i th vector within the array and the j th class within the probability distribution vector.

Re-ID methods incorporating verification loss [2, 111, 200] (Section 2.3.5) may instead utilise Binary Cross-Entropy (BCE), a variant of CCE which computes the cross-entropy when only two classes are present. This verification task involves a binary decision consisting of whether or not multiple input images represent the same identity. Given a ground-truth vector Y and predicted vector \hat{Y} , both consisting of N samples, the Binary Cross-Entropy (BCE) between the vectors is defined as:

$$\epsilon_{BCE}(Y, \hat{Y}) = -\frac{1}{N} \sum_{i=0}^{N-1} Y_i \log(\hat{Y}_i) + (1 - Y_i) \log(1 - \hat{Y}_i). \quad (2.5)$$

In general, given a loss function, ψ , which outputs an error ϵ , the task of a neural network is to minimise the error ϵ over the validation set. Before training, network weights are initialised randomly. Training is usually carried out on mini-batches of size k , where the error rate of a given batch, ϵ_b is calculated by averaging

the sum of the error from each individual sample:

$$\epsilon_b = \frac{1}{k} \sum_{i=0}^{k-1} \epsilon_i \quad (2.6)$$

The information gained by evaluating the performance of the network with the loss function ψ is used to increase the performance of the network by computing the gradient of the error with respect to the weights of the network. This gradient provides information on the influence of each weight on the error ϵ . This process is known as backpropagation. Given a weight w_{ij} connecting neurons i and j , the gradient of the error ϵ_b with respect to the weight w_{ij} can be calculated by:

$$\frac{\partial \epsilon_b}{\partial w_{ij}} = \frac{\partial \epsilon_b}{\partial z_j} \frac{\partial z_j}{\partial x_j} \frac{\partial x_j}{\partial w_{ij}} \quad (2.7)$$

where z_j is the output of neuron j , and x_j is the weighted input of neuron j [157]. Optimisers, such as RMSProp [177], are then used to perform learning and updating of weights, where a given weight w_{ij} connecting neurons i and j at time t is updated by:

$$\begin{aligned} \mu(t) &= v\mu(t-1) + (1-v) \left[\frac{\partial \epsilon_b}{\partial w_{ij}(t)} \right]^2 \\ w_{ij}(t) &= w_{ij}(t) - \frac{\eta}{\sqrt{\mu(t)}} \frac{\partial \epsilon_b}{\partial w_{ij}(t)} \end{aligned} \quad (2.8)$$

where μ_t computes a moving average of the squared gradients at time t , v is a parameter for weighting the contribution of the moving average at the previous time step versus the gradient of the error with respect to the weight at time t , and η is the initial learning rate [96]. As such, the weights are updated in a way which minimises the error ϵ . Hence, the network should output more accurate predictions aligned better with ground-truth values.

Following completion of the training stage, the model is presented with unseen data which forms the testing set. This gives an estimate of the models ability to generalise, as the testing set is not used to train the model. Evaluation is typically carried out by utilising an evaluation function such as *precision* and *recall* (Section 2.3.7), which measures the performance of a model compared to ground-truth data.

2.3.2 Layers used commonly in Convolutional Neural Networks

Convolutional Layers

Artificial neural networks consist of a series of hierarchical layers connected by weights and biases. Images are passed layer-by-layer to generate feature maps. A common layer present within CNNs is a convolutional layer, where a kernel is convolved with the input to produce a feature map. Given a two-dimensional image I and a two-dimensional, the output of the 2D convolution operation, S , can be defined as:

$$S(i, j) = (I * K)(i, j) = \sum_m \sum_n I(m, n)K(i - m, j - n), \quad (2.9)$$

where m and n represent summation over the image dimensions [55]. A simple example can be seen in Figure 2.1. As convolution is commutative, it can also be written as:

$$S(i, j) = (K * I)(i, j) = \sum_m \sum_n I(i - m, j - n)K(m, n), \quad (2.10)$$

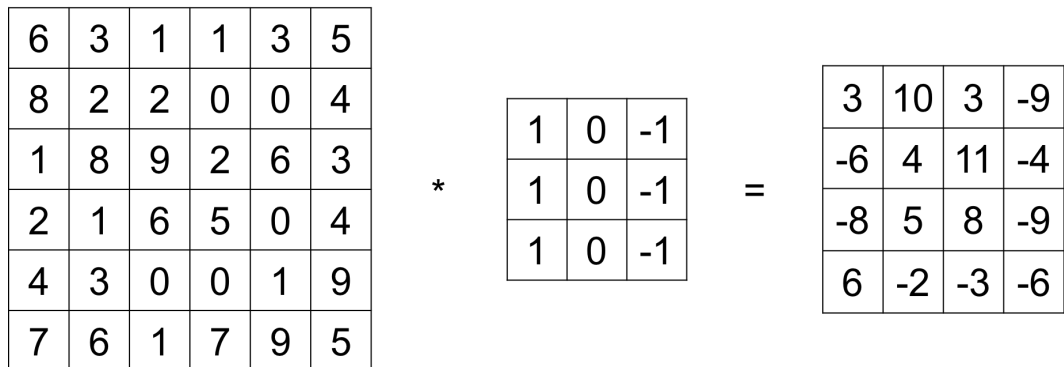


Figure 2.1: An example of the convolution operation applied on a 6×6 matrix with a 3×3 filter.

Thus, the goal of training a deep CNN is to learn a set of kernels which are capable of detecting the presence of certain visual characteristics within an image. Kernels at the input of the network will learn to detect the presence of certain lower-level features such as lines and edges, whereas kernels at the output of a network will learn to detect the presence of more higher-level, field-specific visual characteristics of an image. The output of a convolutional layer can be written as:

$$\kappa_1 = \Phi(w * \kappa_0 + \beta) \quad (2.11)$$

where κ_0 is the input feature map, κ_1 is the output feature map, w is the weights, b is the bias, $*$ is the convolution operation, and Φ is an activation function such as the sigmoid, softmax [16] or ReLU. An example of the 3×3 filter from Figure 2.1 applied to an image can be seen in Figure 2.2.



Figure 2.2: An example of the 3×3 filter in Figure 2.1 applied to an image. It can be observed that the filter highlights vertical edges within the image. In CNNs, the weights of the filter are learned by backpropagation.

Pooling Layers

Pooling layers are used to downsample feature maps by a given factor, and is useful for many reasons. Firstly, pooling reduces the spatial dimensionality of the feature maps, making the features much easier and more computationally efficient to process. In addition, pooling reduces the complexity of the network, which also reduces the number of trainable parameters and hence reduces the time and memory needed for the network to converge. Pooling operations also bring a degree of spatial invariance to a CNN, allowing the network to recognise certain visual characteristics of an image regardless of the physical location of these characteristics. Within the field of Re-ID, spatial invariance is highly important in a CNN given the pose variation of people within Re-ID images. The two main pooling operations implemented within CNNs are max-pooling (Figure 2.3) and average-pooling (Figure 2.4).

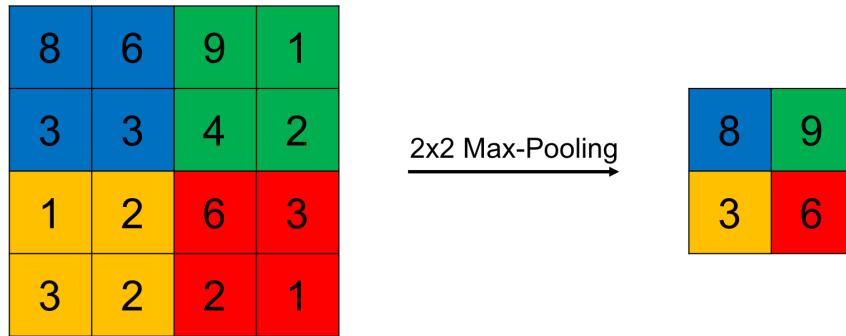


Figure 2.3: An example of max-pooling used to downsample a feature map. Within each pooling region, represented by a distinct colour, the maximum value is taken to form the corresponding value in the output feature map.

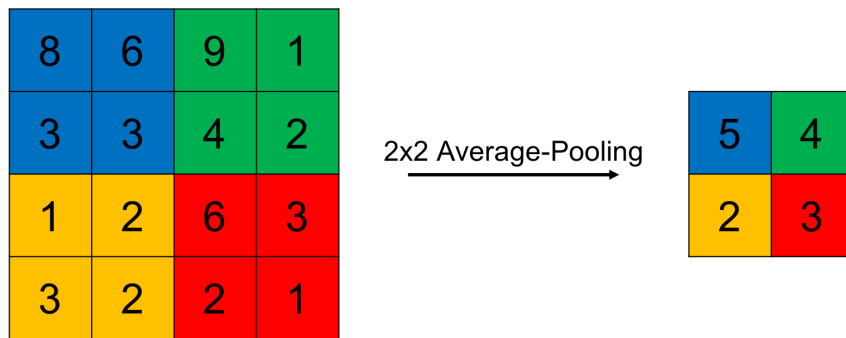


Figure 2.4: An example of average-pooling used to downsample a feature map. Within each pooling region, represented by a distinct colour, the average value of all values is taken to form the corresponding value in the output feature map.

Fully-Connected Layers

Once the dimensionality of the input has sufficiently decreased in size, the output is often passed to one or more fully-connected layers. The layers are named fully-connected as the units in each layer are connected to every unit in the previous layer. Figure 2.5 demonstrates the layout of a simple network containing only fully-connected layers.

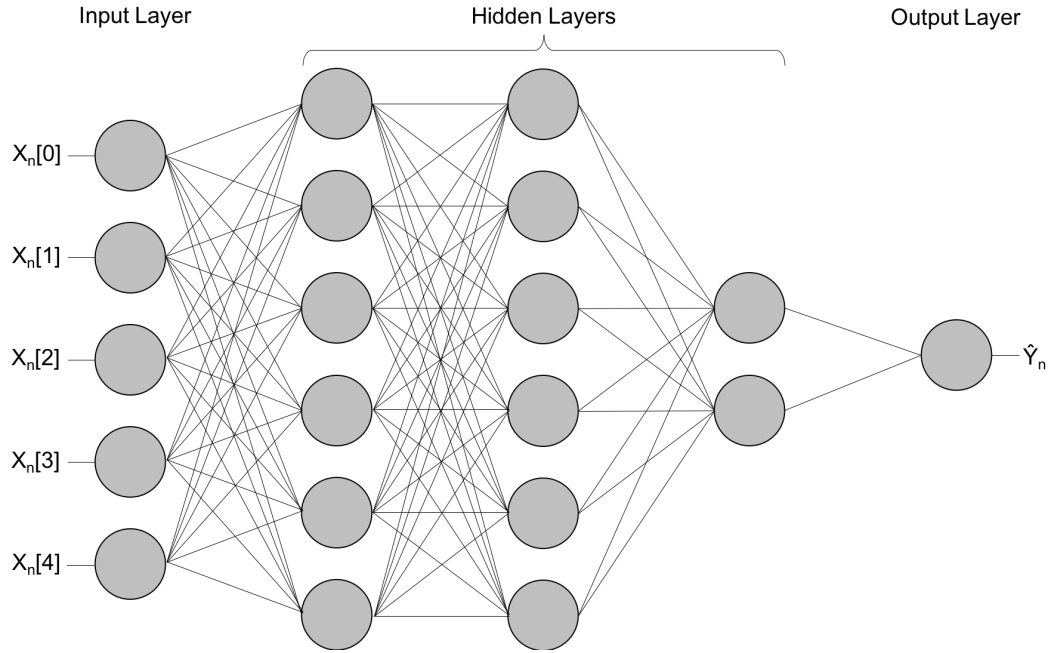


Figure 2.5: A simple network containing only fully-connected layers. The network takes in input X_n of length 5, and predicts \hat{Y}_n of length 1.

These layers formalise meaningful information from any earlier layers. For example, for the task of classification, a fully-connected layer is used in combination with a softmax activation function (Equation 2.3). The number of units within the final fully-connected layer is equal to the number of classes present within the training set. Given a set of z classes $(c_0, c_1, \dots, c_{z-1})$, the softmax activation function computes a series of probabilities $(p_0, p_1, \dots, p_{z-1})$, where each value represents the probability that the input image is a member of the corresponding class. However, unlike the sigmoid activation function, the sum of all probabilities equals one. Within the context of Re-ID, classes are typically identities, with each unit in the final fully-connected layer representing one identity [108, 112, 203, 206]. As such, the identity corresponding to the unit with the maximum assigned probability in the final fully-connected layer is taken as the predicted identity. Alternative activation functions can be used in co-operation with fully-connected layers, such as sigmoid and ReLU.

Often, layers of a CNN which are in between the input and output layers are referred to as hidden layers, and are typically a combination of convolutional and pooling layers. Within Re-ID, Fully-Connected Layers are often found at the end of a network, being used to map the feature maps to a series of units, with each unit representing a specific person identity. Figure 2.6 shows a typical example of a CNN network trained to predict the identity of a person.

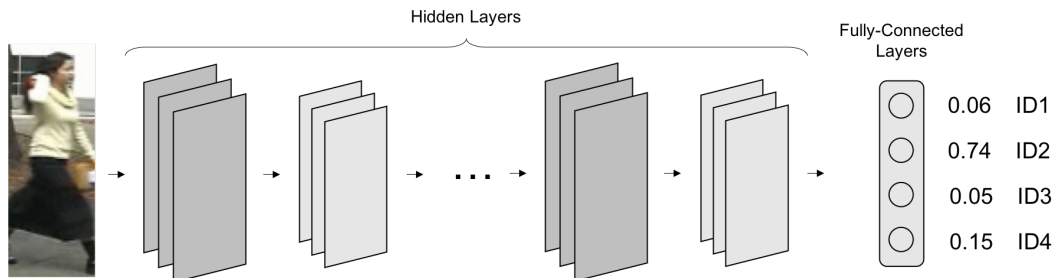


Figure 2.6: A typical use-case for a CNN within the field of Re-ID. A pedestrian image from the VIPeR [59] data set is passed as input to a CNN, and is passed through a series of hidden layers to create a series of feature maps. Once the final feature map has been computed, it is passed to a fully-connected layer, where each unit within the fully-connected layer represents an identity. In this example, the network has predicted with 74% confidence that the identity of the person present in the input image is ID2.

Similarities are present between fully-connected layers and convolutional layers. Assuming a convolutional layer with a kernel size of greater than 1×1 pixel, each unit within a convolutional layer will receive input from a number of local units within the previous layer. This is similar to fully-connected layers, however, fully-connected layers instead receive input from *all* units within the previous layer. Hence, the *receptive field* of a fully-connected layer is typically much larger than the receptive field of a convolutional layer. A downside to this is that fully-connected layers discard information relating to the spatial structure of the input data, treating input pixels which are a large distance from each other equal to those which are a much closer distance to each other. However, the reduced receptive field of convolutional layers leads to better identification of local features within an input, making these layers better suited to computer vision classification tasks.

Additionally, whilst fully-connected layers can be used to learn features to classify image data, the large number of weights within each layer cause them to be impractical when dealing with high-dimensional image data. For example, when using a fully-connected layer with 32 units, and an image with dimensions $224 \times 224 \times 3$, the resulting fully-connected layer would contain $224 \times 224 \times 3 \times 32 = 4,816,896$ trainable weights. Comparatively, consider a convolutional layer which takes as input a $224 \times 224 \times 3$ image, and convolves with 32 kernels of size $5 \times 5 \times 3$ pixels each, with a stride of 3, where the stride is defined as the number of pixels which the kernel moves across the input image per convolutional operation. The convolutional layer would be of size $74 \times 74 \times 32$ units. Therefore, without any form of parameter sharing, each of the $74 \times 74 \times 32$ units would consist of $5 \times 5 \times 3$ weights, resulting in $74 \times 74 \times 32 \times 5 \times 5 \times 3 = 13,142,400$ weights. However, as

convolutional layers would apply the same kernel to all of the units within the same 74×74 layer via parameter sharing, this reduces the number of trainable weights down to $32 \times 5 \times 5 \times 3 = 2,400$ trainable weights. Hence, the use of convolutional layers significantly decreases the number of trainable weights within the network, increasing network efficiency and preventing overfitting [33].

2.3.3 Activation Functions

Activation functions are an abstract representation of the action potential of a cell [72], and map the output of a layer into a desired range, such as between 0 and 1. This introduces non-linearities into a CNN, increasing a networks capability to learn more complex mappings. One of the most widely-used activation functions is the sigmoid activation function, $\sigma_{sigmoid}$, which can be defined as:

$$\hat{x} = \sigma_{sigmoid}(x) = \frac{1}{1 + e^{-x}} \quad (2.12)$$

where x is the input feature map and \hat{x} is the output feature map. The use of a sigmoid activation function maps the output to within the range of 0 and 1, making the sigmoid activation function highly useful for expressing probability. However, activation functions such as sigmoid can suffer significantly from the *vanishing gradient problem* [15], where for a layer input x , the partial derivative of the cost function with respect to the current weights tends to zero when $|x|$ is large. The outcome of this is that the gradients of the network during training can become so small, that the weights are prevented from changing significantly. Furthermore, sigmoid activation functions are often used for multi-label classification, as they allow the network to calculate the probability that a particular label is true for a given input. As Re-ID involves predicting the identity of an input person, and each person can have only a single identity, sigmoid activation is not utilised within identity prediction. However, sigmoid activation functions can be used within the problem of Re-ID for other purposes, such as attribute prediction [21, 116, 171], where the probability of an input image belonging to each class (attribute) is computed.

Other activation functions, such as ReLU, suffer less from the vanishing gradient problem by enforcing a gradient of 0 or 1 dependent on the input [137]. ReLU can be written as:

$$\hat{x} = \max(0, x). \quad (2.13)$$

Therefore, any input values less than zero are replaced with zero. The advantages of ReLU are that it is more computationally efficient to compute, whilst also minimising the negative impacts of the vanishing gradient and exploding gradient

problems [143]. Furthermore, networks utilising ReLU activation functions have been found to train significantly faster than methods using alternative activation functions [95]. However, ReLU suffers from the *dying ReLU* problem [32], where weights are updated in such a way that the neuron never activates. Hence, the gradient flowing through the neuron will remain at zero. Attempts have been made to resolve this by methods such as the *Leaky ReLU* [207], where instead of setting the output of a ReLU to be a zero if $x < 0$, it is instead set to a very small value.

2.3.4 Training

Hyperparameters

A neural network consists of a multitude of parameters such as weights and biases which are optimised during training to produce the most accurate prediction results when given an unseen test sample. However, neural networks also contain a set of parameters, known as *hyperparameters*, which are predefined during the design of the network architecture rather than learnt during training.

There are a wide variety of different types of hyperparameters within a neural network. Some relate to the size and architecture of a network, such as the number of hidden layers, as well as the number of units within each layer. In addition, other hyperparameters define how a network trains, such as the learning rate or batch size. Clearly some hyper-parameters are dependent on others, such as the number of units within a particular hidden layer being dependent on the number of hidden layers [30].

However, for a given neural network, there is no single set of hyperparameters which works for all data sets. van Rijn and Hutter [182] pose two questions when choosing hyperparameters:

- Which hyperparameters yield the greatest empirical performance?
- What are the most optimal values for these hyperparameters, to achieve the greatest performance?

The authors found that whilst the same hyperparameters were generally important across data sets, different hyperparameters were important for different model architectures. As such, hyperparameters are generally estimated by repeated training on a given data set with a variety of different values, observing the performance of each network following each trial.

Overfitting and Underfitting

If the performance on a testing set is significantly worse than the equivalent performance on a training set, the network is said to have overfitted [99]. This is related to the problem of generalisation, where a model trained on one subset of data may not perform well on another.

Overfitting is generally avoided by incorporating regularization into the training stage. One of the most common regularization techniques is to apply ℓ_1 or ℓ_2 regularization [48], which penalises complex models by adding an additional term to the loss function. Utilising ℓ_1 regularization, given a loss function ψ , the updated loss function $\hat{\psi}$ is computed by:

$$\hat{\psi} = \psi + \lambda \left(\sum_{i=0}^{I-1} |w_i| \right) \quad (2.14)$$

where λ is a regularization term which controls the importance of the regularization, I is the number of weights within the network, and $\mathbf{w} = \{w_0, w_1, \dots, w_{I-1}\}$ are the weights. Similarly, utilising ℓ_2 regularization adds to the loss function the sum of the squared elements of the weight matrix, by:

$$\hat{\psi} = \psi + \lambda \left(\sum_{i=0}^{I-1} w_i^2 \right) \quad (2.15)$$

Thus, overly complex models which have a greater tendency to overfit the training data are penalised further than simpler models. Another solution to the problem of overfitting is to apply Dropout [69]. Dropout prevents overfitting by randomly excluding individual units within a layer at a prespecified rate [169]. By randomly ignoring a percentage of the units within a layer, dropout introduces noise into the training process. As the decision for which units to drop is random and carried out on an epoch-by-epoch basis, the units which are dropped differ every epoch, and therefore make it harder to overfit given the variability of data passed through the network. Furthermore, overfitting can be caused by training the network for too long on the training data [50]. To prevent this, early stopping is used to ensure that the network stops training once the networks performance stops increasing. Therefore, the weights which give the best prediction performance are preserved.

A similar problem is underfitting. Underfitting occurs when the network is unable to learn a sufficient mapping between the input and output variables, and is therefore unable to achieve good accuracy. In this case, the model is said to have underfitted. Underfitting can have many causes. Firstly, it is possible that hyperparameters are set to inappropriate values, such as the number of units within

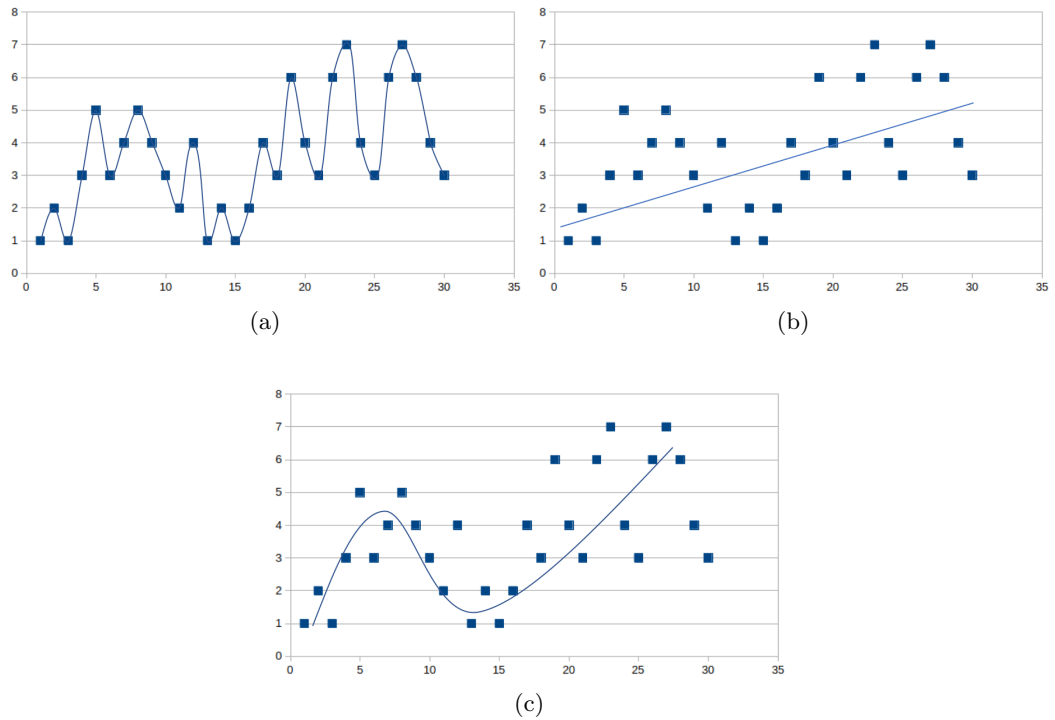


Figure 2.7: Examples of (a) overfitting, (b) underfitting and (c) achieving a good fit on a data set.

a fully-connected layer being too small in relation to the complexity of the problem, and therefore the network may be unable to learn an appropriate mapping [141]. Additionally, underfitting may be caused simply by the lack of training data necessary for the network to learn a high-performing mapping between the input and output data. As such, solutions to the problem of underfitting can be to experiment with hyperparameters, as discussed in Section 2.3.4, and to ensure the quantity of training data is sufficient for training the network. Figure 2.7 shows examples of overfitting, underfitting and achieving a good fit on a data set.

2.3.5 Training Strategies

There are multiple different methods to train a deep CNN. The ID Classification Loss approach can be defined as learning features optimal to the task of predicting the identity of a particular person. As a classification problem, this means that each identity is its own class, leading to a typically high number of classes needing to be learned. This is in contrast to Verification Loss, where the network seeks not to predict the identity of a given person image, but instead to verify whether or not two person images represent the same identity. Furthermore, Triplet Loss is a learning process where a triplet consisting of an anchor, positive match and negative match

are passed as input to a network. The aim of this kind of network is to minimise the distance between the anchor and positive match, whilst simultaneously maximising the distance between the anchor and the negative match.

Verification Loss

Initial uses of deep learning in Re-ID generally incorporated Verification Loss, which is demonstrated in Figure 2.8. The first use of deep learning techniques for Re-ID was proposed by Li et al. [111], where the authors define their own network architecture named Filter Pairing Neural Network (FPNN). The authors propose the inclusion of a Patch Matching layer, which compares two sets of features learnt from two different input images, and from this calculates a set of displacement matrices. This layer is followed by a Maxout-Grouping layer, where only the displacement matrices with the highest activations are passed to the following layers. The subsequent layers are further convolutional and max-pooling layers, followed by a fully-connected layer which results in the final feature descriptor for each of the input images. The final layer uses a softmax function to determine whether the two input images represent the same identity or otherwise. This was extended by Ahmed et al. [2], where the authors propose learning cross-input neighborhood difference features, which consists of comparing features from a particular region in one input image with the features obtained from neighbouring locations in the second input image. The justification for this is to add robustness to positional variation present between the two input images. The output of this layer produces an approximate relationship between the features extracted from the two input images. The authors additionally propose an additional novel layer which summarises these neighbourhood difference features into a smaller feature descriptor, which is then passed through a further Convolutional and Max-Pooling layer. Finally, two fully-connected layers and a softmax function are used to determine whether or not the two input images belong to the same identity.

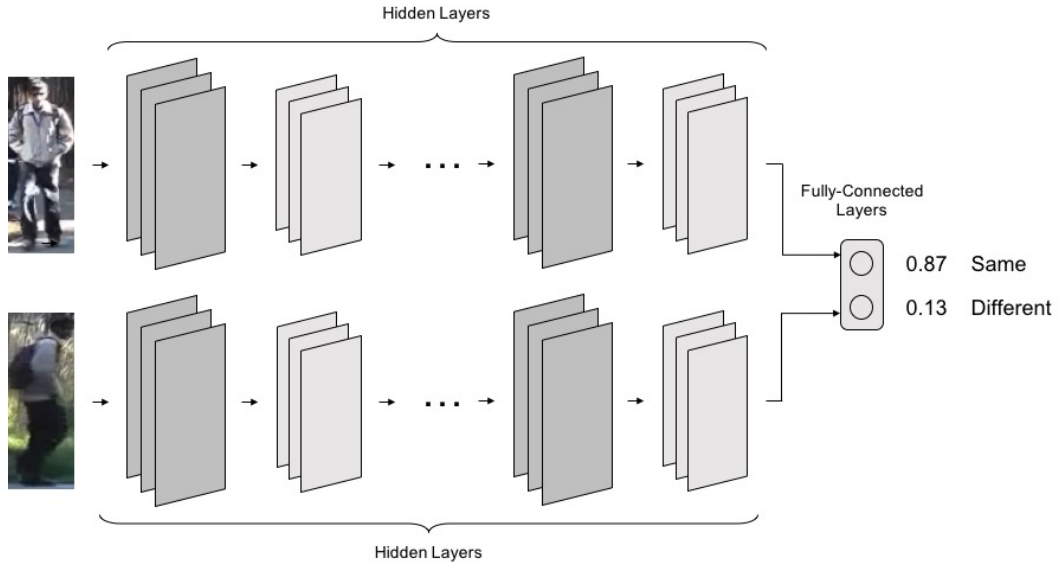


Figure 2.8: An example of a network trained using Verification Loss. The network is trained to predict whether or not a given pair of images represent the same identity or otherwise. The example images are taken from the VIPeR [59] data set.

Wu et al. [200] expands on the work by Ahmed et al. [2], also incorporating a cross-input neighbourhood differences layer to learn the relationship of patches between distinct Re-ID images. The authors argue that a deeper model than that used within [2] is necessary for higher matching rates due to the significant spatial misalignment present within Re-ID data sets, and state that the deeper network would be much better suited to handle this misalignment. The authors add a series of convolutional layers with 3×3 pixel convolutional filters, which capture non-linear relationships between patches.

ID Classification Loss

More recent work has focused on ID Classification Loss, which is demonstrated in Figure 2.9. ID Classification Loss considers each identity as its own class, and hence requires a larger number of samples per identity compared to when using Verification Loss [221]. Xiao et al. [206] discusses the problem of generalisation, where a model trained on one data set cannot be effective when evaluated on another. They resolve this by proposing a three stage training method. First, images from a variety of different data sets are combined into one larger data set and trained with standard dropout. Secondly, the dropout is replaced with the author’s proposed Domain Guided Dropout (DGD), which assigns each neuron within a CNN a dropout rate for each data set, according to its effectiveness for that data set. Finally, the CNN is fine-tuned on each data set separately. Li et al. [108] propose Multi-Scale Context-Aware

Network (MSCAN), which consists of a series of multi-scale convolutions in each layer. Each person image is divided into three parts, and the whole image as well as each part is passed through the MSCAN. Finally, the whole image and body part learnt representations are integrated through concatenation, and cross entropy loss is used to predict the identity of the person. Li et al. [112] proposes Jointly Learning Multi-Loss (JLML), where the network aims to jointly learn both local and global features simultaneously by finding correlations between the two. A two-branch CNN is proposed, with one branch focusing on global features whilst the other focuses on local features. The authors enforce a separate loss function per branch to learn independent discriminative features whilst enforcing the same ID label constraint. For testing, the feature descriptors from each branch are concatenated to create the final feature descriptor to represent the entire image.

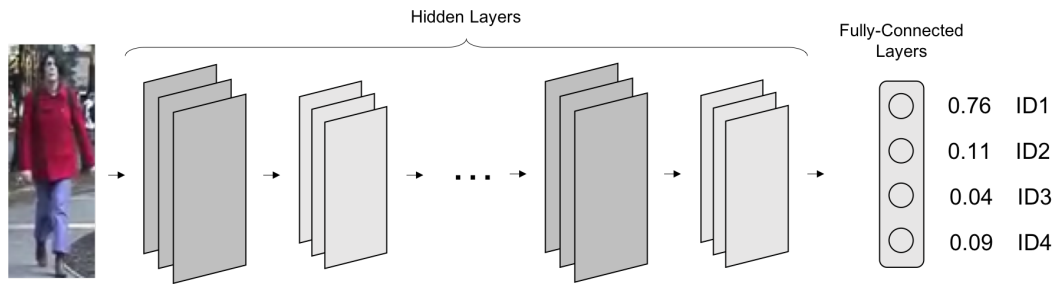


Figure 2.9: An example of a network trained using ID Classification Loss. The network is trained to predict the identity of a given image. The example image is taken from the VIPeR [59] data set.

Wu et al. [203] propose combining both traditional hand-crafted features with deep features. The authors based their hand-crafted features on the ELF feature [58], as improved by [226, 227], which consist of RGB, HSV and YCbCr colour histograms as well as 8 Gabor filters and 13 Schmid filters being extracted from 6 horizontal stripes. However, they modify ELF to instead extract 16-dimensional RGB, HSV, LAB, XYZ, YCbCr and NTSC colour histograms, as well as 16-dimensional Gabor, Schmid and LBP texture histograms from 16 horizontal stripes. All histograms are ℓ_1 normalised, and then concatenated. Regarding the deep features, the authors propose five convolutional layers, which are each followed by a pooling layer and a Local Response Normalization (LRN) [95], except for the third layer. The output of the fifth pooling layer is utilised as the deep feature descriptor. The deep feature and hand-crafted feature are then fused using a further fully-connected layer, with a softmax function used to predict the identity of the person.

Combination of Verification Loss and ID Classification Loss

Often, network architectures can incorporate both verification and ID classification loss within a single network. These networks simultaneously learn features suitable for both multi-class recognition of a given persons identity, whilst also being able to determine whether a given pair of images belong to the same identity or otherwise. Geng et al. [51] proposes a network architecture which utilises both verification and ID classification loss. However, the authors argue that given two feature maps extracted from a Re-ID image pair, using different dropout maps on each feature map would not be optimal for computing the verification loss, as some differences between the two feature maps could be explained simply due to the randomness of the dropout masks rather than the visual appearance of the people present within the Re-ID images. Therefore, the authors propose a Loss Specific Dropout Unit, which ensures the same dropout map is applied to both feature maps extracted from a given image pair. Following the Loss Specific Dropout Unit, the resulting feature map pair are passed to a verification loss layer, and each feature map of the pair is individually passed to an ID classification loss layer. Zheng et al. [229] propose a network where given a pair of images, each image is passed through a separate branch of a CaffeNet [95], VGG-16 [167] or ResNet-50 [66] network, where both branches share weights with the other. Each branch possesses an ID classification loss layer, which uses a softmax function to predict the identity of the input image. The final feature maps of each branch are then taken and compared using the authors proposed Square Layer, which takes two feature maps and subtracts one feature map from the other, and performs a square operation element-wisely. Finally, the output feature map is used as input to the verification loss layer, which uses a softmax loss function to predict whether or not the two input images represent the same identity. Zhong et al. [230] proposes an architecture which also utilises both verification loss and ID classification loss. Firstly, each image of an image pair are passed through a feature extraction network named Feature Aggregation Network (FAN), which is based on a ResNet-50 [66] network architecture. FAN extracts features maps from different layers in the network and provides an element-wise sum to produce the fused feature map. Afterwards, for the verification branch, the feature maps are passed to a Recurrent Comparative Network (RCN), which compares the appearance of images that form an image pair by using a combination of attention and recurrent networks. The attention component weights discriminatively significant regions of the feature map, whilst the recurrent networks aggregate the discriminatively significant regions within the image pair. The output of the RCN is then used as input to the verification loss layer, computing whether or not the two input images are of the same person. For the ID classification branch, the feature maps extracted from

the FAN are pooled using a global average pooling operation, and then passed to a fully-connected layer with softmax loss to predict the probability that the image belongs to each class.

Triplet Loss

Training a network using triplet loss is highly useful within the field of Re-ID, where the task is to ensure the smallest possible distance between feature descriptors extracted from those of the same identity. Methods which utilise triplet loss take as input three images - a probe image, and image with the same identity as the probe image, and an image with a different identity to the probe image. The task is then to minimise the distance between the images with the same identity, whilst maximising the distance between the images with different identities. Figure 2.10 illustrates this process.

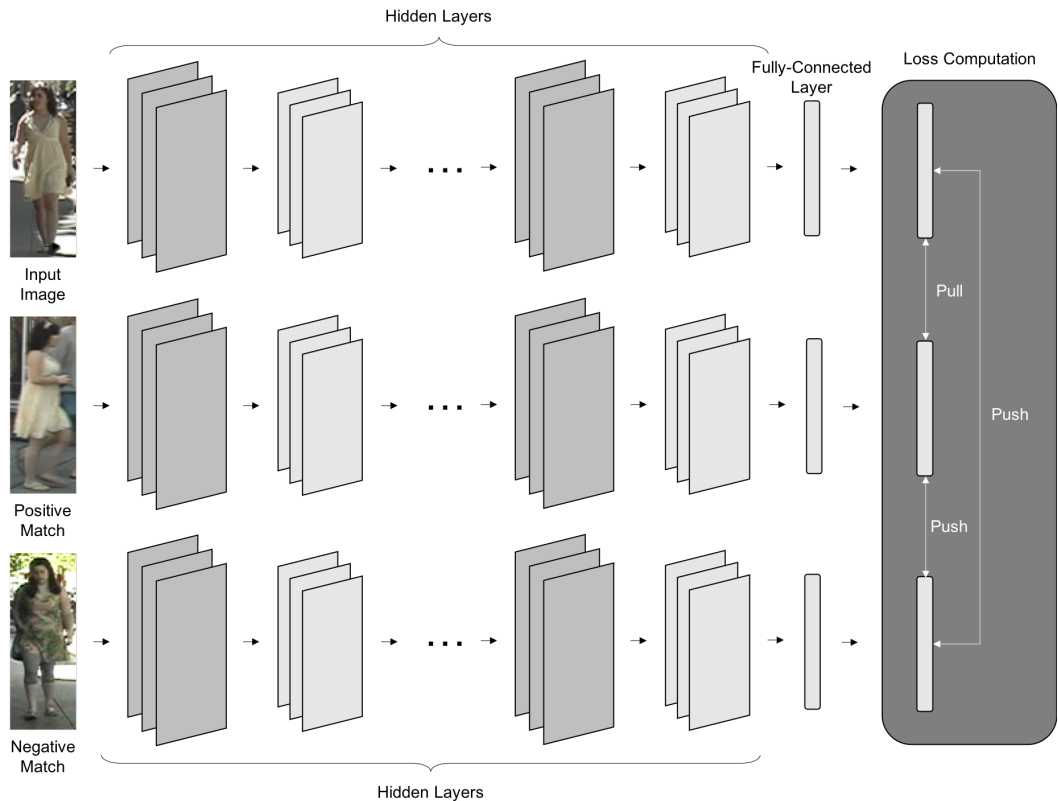


Figure 2.10: An example of a deep neural network utilising triplet loss. Three images are passed to the network, an *input image*, a *positive match* image with the same identity as the input image, and a *negative match* with a different identity to the input and positive match images. The network *pulls* the output of the input and positive match to be closer, by minimising the distance between the two within the output feature space. Furthermore, the network *pushes* the input and positive match away from the negative match, by maximising the distance between the different identities within the output feature space.

Multiple methods have employed this technique. Khamis et al. [90] jointly learn a projection to a appearance-attribute subspace. The authors argue that the presence of the attribute information helps their method achieve some invariance to illumination variation. Their method jointly optimises a ranking loss for Re-ID matching, as well as attribute classification. Cheng et al. [28] proposed a method where both global image features as well as local body part image features are concatenated and passed to a triplet loss function. Furthermore, Cheng et al. [28] argue that the typical triplet loss training strategy, where the distance between the anchor and positive input must be smaller than the distance between the anchor and negative input by a pre-defined margin, is not optimal for Re-ID, due to the possibility that all input of a given identity will form a large cluster with a large average intra-class distance between instances. Consequently, the authors propose a

further constraint ensuring that the distance between the anchor and positive input must be below a certain threshold, and demonstrate that using this improved triplet loss function can improve matching rates by up to 4%. Chen et al. [27] propose a network architecture which jointly optimises both a binary classification loss, as well as ranking loss. They utilise triplet loss for the ranking part of their network, whilst utilising binary softmax for the classification component. The authors argue that the combination of both ranking utilising triplet loss as well as classification using softmax in a joint network takes advantage of the complementary of both methods. In [118], the authors propose the end-to-end Comparative Attention Network (CAN), which learns which parts of images are more discriminative by building an attention-based model, where the model is optimised using triplet loss. Zhao et al. [214] proposes constructing a feature descriptor using a concatenation of feature maps from multiple different layers from within a network. This combines shallow layer information with deeper layer information, permitting different types of visual characteristics to be captured by each component which makes up the final feature descriptor. These multilevel features are then used as part of a triplet loss training strategy.

2.3.6 Transfer Learning

Training a neural network from scratch can take significant time, as well as computational resources. Furthermore, a considerable amount of training data is required to build a high-performing neural network. To decrease the resources necessary for training a network from scratch, it is common for these networks to be not initialised with random weights, but with weights pre-trained on an similar problem. This is analogous to how a human can use knowledge from one task to learn another [179].

An example of pre-trained weights which are often transferred to other, related domains, such as Re-ID, are weights trained on the ImageNet data set for object recognition [38]. The ImageNet data set contains over one million images with bounding boxes, representing over 1000 different classes. Furthermore, several of the classes found within ImageNet are visually similar, such as six different varieties of spider. This forces the network to learn discriminative, robust features to differentiate between visually similar classes [81]. Notably, due to ImageNet's large number of classes, it contains knowledge which is appropriate for a vast number of domains. For example, the ImageNet data set contains classes such as *sweatshirt*, *jean* and *umbrella*. This demonstrates similarity between the task for which ImageNet has been trained, and the attribute prediction task as described in Chapter 5. By initialising the weights of an attribute prediction network with ImageNet's pre-trained weights, this gives a head-start on training, increasing performance whilst decreasing the time required to train the network. Furthermore, it also potentially reduces the chance of

the network overfitting.

Following the initialisation of a network with pre-trained weights from another domain, the next step is to alter the architecture of the network to fit the target domain. This typically requires the replacement of the final, fully-connected layer to contain a number of units equal to the number of classes within the target domain [165]. However, given the difference in domain, the pre-trained weights for this final layer cannot be utilised for the target domain. Therefore, the network is then further trained on one or more data sets from the target domain, in a process known as fine-tuning or refinement [81]. Fine-tuning a network maintains the knowledge obtained during initial training on the source domain data set, whilst optimising the weights for the target domain, and training any additional layers such as the fully-connected layer representing the target domain’s class labels. It has been demonstrated that utilising transfer learning to transfer knowledge from one domain to another increases prediction results [51, 149, 205].

2.3.7 Evaluation

The performance of a deep method can be evaluated in multiple ways. The performance is evaluated during training in order to alter the weights of the network through backpropagation. However, following completion of the training stage, it is important to evaluate the data on unseen data to ensure that the network can generalise.

It is common for a network to be evaluated using a different method to that which was used to optimize the network during the training stage. For example, the BCE function typically used to train Re-ID networks optimizes a networks ability to predict the identity of a given individual, whereas the rank- n evaluation method used to evaluate unseen Re-ID images focuses on the networks ability to predict the identity of a given individual within n guesses.

During post-training evaluation, each instance of unseen data is passed through the trained network, and a predicted value is obtained. One of the most common evaluation metrics used to measure the performance of a neural network is *accuracy*. Given the number of True Positives (TPs), False Positives (FPs), True Negatives (TNs) and False Negatives (FNs), the accuracy is calculated as:

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (2.16)$$

However, given an unbalanced data set, evaluating according to the accuracy metric can be unhelpful. Consider a network which predicts whether or not a group of individuals will be diagnosed with a serious disease over the next year. On the

assumption that the overwhelming majority of people will not be diagnosed with a serious disease over the next year, the network can achieve a very high accuracy simply by classifying each instance as negative.

One of the ways to overcome the bias is to choose a more suitable evaluation metric given the problem. For example, *precision* is defined as:

$$precision = \frac{TP}{TP + FP}. \quad (2.17)$$

Precision calculates the percentage of all true positives against all values predicted as positive by the network, i.e. the percentage of those identified positive by the network that were truly positive. Another evaluation metric, *recall*, is defined as:

$$recall = \frac{TP}{TP + FN} \quad (2.18)$$

Recall calculates the percentage of true positives against all values which are positive in reality, i.e. the percentage of truly positive values correctly classified. By using precision and recall, even given an unbalanced data set, the network would be unable to simply predict the most common class to achieve a high evaluation score. Therefore, it is crucial when building a neural network to choose an evaluation metric which conveys the networks ability to output informative predictions, rather than those which score the best results for an arbitrary evaluation metric.

Within the field of Re-ID, the rank- n evaluation metric (Section 2.6) is typically utilised to evaluate the performance of each method. This metric measures the percentage of probe images which are matched with their corresponding gallery images within n guesses. However, evaluation of a skeleton or pose prediction model requires an evaluation method able to measure the differences between a set of continuous variables. For these problems, the Mean Squared Error [10, 97] and Root Mean Squared Error [5, 74, 183] are utilised, demonstrating the scale at which predicted skeletons and poses vary from the ground-truth.

2.3.8 Cross-Validation

If the data sets used are small in size, the training and testing sets can contain heavy bias if the distribution of class labels or visual characteristics are not well spread between the training and testing sets. Then, if a method such as hold-out validation is used to evaluate the performance of a model, where a data set is divided into a training set and testing set and evaluation carried out only once, the evaluated performance of a network may not accurately reflect its true ability to generalise.

In order to minimise the negative effects of bias, cross-validation is used to

conduct testing more representative of the performance of the network. There are two main ways to undertake cross-validation. The first is known as *k-fold cross validation* [94], where a data set, D , is randomly partitioned into k mutually distinct subsets $(D_0, D_1, \dots, D_{k-1})$ of approximately equal size. One of the subsets is then used as the test set, with the remaining $k-1$ subsets being used for training. The process is repeated k times, with a different subset being used as the testing set each time. The final testing score is then computed by averaging the score of each of the k test sets.

The second is known as *repeated hold-out validation* [91], and involves dividing the data set into training and testing sets based on a pre-defined percentage split, in the same way as hold-out validation. However, similarly to k -fold cross validation, experimentation using repeated hold-out validation is carried out multiple times, with the final testing score calculated by averaging the score from each individual testing split.

2.4 Attribute Learning

Whilst traditional Re-ID methods have related to describing person images with low-level features such as colour and texture, these feature types can be heavily influenced by variations in illumination or other visual characteristics. However, characteristics such as age, gender and general colour of shirt, are significantly more invariant to these changes. Thus, several pieces of the literature have researched how best to incorporate such characteristics, typically referred to as *attributes*, into the Re-ID pipeline.



Figure 2.11: Example attributes and corresponding positive and negative examples. Images are taken from the VIPeR [59] data set with attribute labellings taken from the PETA [39] data set.

Layne et al. [101] define a mid-level attribute as being a physical characteristic that is unambiguous in interpretation. Fifteen attributes are chosen, namely *shorts*, *skirt*, *sandals*, *backpack*, *jeans*, *logo*, *v-neck*, *open-outwear*, *stripes*, *sunglasses*, *headphones*, *long-hair*, *short-hair*, *gender* and *carryingobject*. As some of these attributes will only be present on certain parts of a person’s body, the authors extract a 464-dimensional feature descriptor from six equal-sized stripes, resulting in a 2784-dimensional feature descriptor consisting of colour and texture information. The authors then train a Support Vector Machine (SVM) [161] to detect the presence of the fifteen attributes. Given that some attributes will be more reliable than others, due to their prevalence within the imbalanced data as well as how useful they are for discriminating between different individuals, the authors learn a weighted ℓ_2 -norm distance metric which will weight attributes according to their usefulness. The

authors found that the highest Re-ID matching results could be obtained when the attribute features were combined with low-level SDALF [43] features for matching.

Khamis et al. [90] also combine attribute features with traditional hand-crafted features. The authors learn a distance metric which learns a discriminative projection in a joint appearance-attribute subspace. The authors optimise the ranking loss and attribute classification loss and by this, achieve some invariance to illumination and pose, and demonstrating improved matching rates over using just appearance or attribute information. Su et al. [170] utilise the correlation of attributes, such as female and long hair, to allow attributes of the same person between multiple cameras to be embedded into a low rank space. Using a low rank space allows for noisy attributes to be pruned, as well as missing attributes, such as those which are incorrectly labelled by a human annotator, to be rectified. However, such a task is computationally expensive, and therefore the authors incorporate a Multi-Task Learning (MTL) [18] algorithm. By considering Re-ID between multiple cameras as related tasks, the authors are able to use MTL to exploit features and attributes shared across multiple cameras to increase efficiency and learn from multiple cameras simultaneously.

Shi et al. [164] discuss the problem of there not being sufficient training data to train a Re-ID framework using attributes which can produce state-of-the-art results. To counter this issue, the authors propose using two fashion data sets, named Clothing-Attribute [25] and Colourful-Fashion [122]. However, given the large difference in visual characteristics between data sets, training a model using these data sets would typically be useless for use on Re-ID data sets. The authors propose taking a generative model approach based on the Indian Buffet Process (IBP) [60], and exploit attribute features at patch-level rather than image-level. The use of patch-based features is used in combination with Bayesian Adaptation to ensure that the learnt model can output a strong patch-level feature capable of being used within a wide range of different domains, including Re-ID.

More recently, CNNs are being used for attribute detection. In [171], the authors propose a three-stage attribute prediction network. In the first stage, the authors training a Deep CNN to predict 105 attributes using the PETA [39] data set. The second stage involves fine-tuning the model on the MOTChallenge [104] data set, this time training using person ID labels and utilising triplet loss. The final stage uses a combination of all previous training data sets for the final stage of fine-tuning, with the output of this stage being named by the authors as *deep attributes*. The authors extend this work in [174] by dividing the attributes in to a set of 15 types, such as *Age*, *Gender*, *CarryObject* and *HairStyle*. Encoding attribute information in this way ensures contradictory information such as *short hair* and *long hair* cannot

co-exist, by enforcing only one positive attribute per type. The final attribute feature is therefore shortened from length 105 to a set of K attributes belonging to C types, $A = \{A^1, A^2, \dots, A^C\}$, where $A^c = \{a_1^c, a_2^c, \dots, a_{K^c}^c\}$ and $a \in \{0, 1\}$ represents the presence of otherwise of a specific attribute.

Ye et al. [209] propose a body parts-based approach which combines colour, texture and attribute features. First, LOMO [115] features are extracted and used both to contribute to the person image’s feature descriptor, and to train a LIBSVM [20] classifier for each attribute. Furthermore, a Sample-Specific SVM (SSSVM) [213] is used to weight each body part according to its contribution to Re-ID Matching. Following calculations of the weights, the weighted distance between corresponding parts of different images are fused, forming the final distance between two images. Zhao et al. [218] utilises video sequences to improve Re-ID matching rates. Feature descriptors are extracted first from individual video frames, and are then divided into groups of sub-feature descriptors corresponding to specific attributes. These sub-feature descriptors are weighted according to the corresponding confidence of attribute prediction. Following weighting, the feature descriptors extracted from each frame are aggregated across the temporal dimension to produce the final sequence feature descriptor.

2.5 Other Related Works

2.5.1 Distance Metric Learning

Whilst early Re-ID techniques typically used Euclidean or cosine distance as a metric to calculate a distance between two feature descriptors, more recent techniques have utilised Distance Metric Learning to learn a distance function more appropriate for the task of Re-ID. Zheng et al. [226] propose a Probabilistic Relative Distance Comparison (PRDC) model, where the model uses an objective function aims to maximise the probability that two feature descriptors extracted from the same individual are closer together in the feature space than two feature descriptors extracted from different individuals. Also, several works have been proposed [36, 194, 195] based on a Mahalanobis [19] distance metric, and have been applied to the Re-ID problem [110]. The aim of using a Mahalanobis distance metric is to learn a linear transformation such that relevant dimensions in the feature space are emphasized. The Mahalanobis distance between two feature descriptors, $D_M(x_i, x_j) = (x_i - x_j)^T \mathbf{M}(x_i - x_j)$, where x_i and x_j are two feature descriptors of image i and j respectively, and \mathbf{M} is the positive, semi-definite matrix to be learnt. The goal of \mathbf{M} is to ensure that $D_M(x_i, x_j)$ is small should x_i and x_j belong to the same identity, and large otherwise. However, prior works using a Mahalanobis distance metric often struggle from over-fitting

due to lack of regularization, and can suffer from scalability issues. Koestinger et al. [93] proposed KISSME, a means to learn a metric based on equivalence constraints, increasing efficiency and reducing over-fitting. KISSME determines whether or not a pair of feature descriptors are equivalent by means of a likelihood ratio test. The distance between two feature descriptors is thus calculated as $\tau_M^2(\mathbf{f}_i, \mathbf{f}_j) = (\mathbf{f}_i - \mathbf{f}_j)^T (\Sigma_I^{-1} - \Sigma_E^{-1})(\mathbf{f}_i - \mathbf{f}_j)$, where Σ_I is the intra-personal scatter matrix, and Σ_E is the extra-personal scatter matrix. In [115], the authors propose Cross-view Quadratic Discriminant Analysis (XQDA), which extends KISSME. Unlike KISSME, XQDA incorporates both dimensionality reduction and metric learning simultaneously, ensuring the learned subspace is optimal for the task of learning an appropriate distance metric.

2.5.2 Generative Adversarial Networks

Generative Adversarial Networks (GAN) were first proposed by Goodfellow et al. [54], where their network learnt how best to create counterfeit images. The GAN does this by creating a counterfeit image and continuously responding to feedback on the quality of its output and using this feedback to improve moving forwards. GANs consist of two main components: a generator and a discriminator. The generator is responsible for creating the counterfeit images, whilst the discriminator compares the output of the generator with ground truth images, providing feedback to the generator on how to alter its counterfeit images to make them more similar to their ground truth counterparts. The generator takes this feedback on board, and the process repeats.

GANs were used by Zheng et al. [228] to generate additional Re-ID images to increase the number of training samples. As the generated images belong to none of the training classes, the authors propose using Label Smoothing Regularization for Outliers (LSRO), and extension of Label Smoothing Regularization [176], to assign a uniform label distribution to the generated images. As well as increasing the number of training samples and introduce more variations in colour, pose and illuminations, the use of additional generated images discourages the network from over-fitting.

To counter the effects of the significant domain gap between different Re-ID data sets, various methods [192, 231] propose using a GAN to transfer the visual characteristics common within one data set to images within another data set. The authors argue that networks which are trained on one data set cannot effectively be used for evaluating on a different data set due to the stark differences between data sets. As such, the authors train a GAN to make an image from data set A appear as if it was taken as part of data set B . This can theoretically allow a network to more effectively use the training set of data set A to train a network which will be used

for evaluating on data set B , leading to a significantly larger training set.

Qian et al. [149] focus on the problem of pose variation, using a GAN to synthesise images with a normalised pose. The authors define eight canonical poses and synthesise eight new images for each real image. This process increases the size of the training set 8-fold, and allows the network to extract features more representative of the individual, unaffected by pose variation. The authors then train two ResNet-50 [66] networks, the first with the original images, and the second with the eight synthesised images. During the testing stage, the authors generate nine feature descriptors by passing the original image through the first ResNet-50 network, and the eight synthesised images through the second ResNet-50 network, and apply an element-wise maximum operation to all nine feature descriptors to generate the final feature descriptor.

Liu et al. [119] takes a similar approach by increasing the size of the training set by extracting pose information from one data set and synthesising new images using this pose information with a separate data set. To increase the overall appearance of the synthetic images, the authors propose using not only the usual discriminator and generator aspects of a GAN, but also introducing a Guider. The Guider ensures that the synthetic images produced are adapted to the Re-ID problem, that being, that they are sufficiently discriminative between identities by enforcing intra-class samples to be closer to one-another whilst inter-class samples are further. The larger training set is then used to train a ResNet-50 [66] and DenseNet-169 [76] network.

2.6 Metrics

The most common means of evaluating Re-ID methods is the rank- n metric, which is used to show the probability that a match between a query image and an image of the same identity within a gallery set is obtained within n guesses or fewer. For example, the rank-5 rate shows the amount of query individuals correctly matched with their gallery counterparts within five guesses or fewer. Evaluation is carried out using either a *single-shot* or *multi-shot* scenario, where the former involves computing a descriptor using only a single image of each person, whereas the latter involves utilising multiple images of each person.

Rank- n results can be presented graphically with a Cumulative Matching Characteristic (CMC) Curve, which plots the number of guesses on the horizontal axis against the corresponding percentage of matches achieved within that number of guesses or fewer on the vertical axis. Calculating the area under the curve (AUC) is also a commonly used metric, with higher values indicating fewer guesses required to find the matching image within the gallery set.

Furthermore, Mean Average Precision (mAP) is often used as a metric to measure the accuracy of a Re-ID method. Given a set of Q query images, denoted by I_1, I_2, \dots, I_Q , the mAP is calculated by:

$$mAP = \frac{\sum_{q=1}^Q AP(I_q)}{Q}. \quad (2.19)$$

where AP denotes the Average Precision. Given a query image I_q and a ranked list of gallery images of length G , denoted J_1, J_2, \dots, J_G , where J_1 is the image considered most similar to I_q by the Re-ID system, the AP for the query image I_q can be calculated by:

$$AP = \frac{1}{\sum_{k=1}^G r_k} \sum_{k=1}^G r_k \left(\frac{\sum_{l=1}^k r_l}{k} \right) \quad (2.20)$$

where r_k is 1 if image J_k within the ranked gallery set has the same identity as the query image I_q , and is 0 otherwise [180]. Whilst rank- n only evaluates a Re-ID method by finding the index of the closest match for each query image in the ranked gallery set, mAP calculates the AP for each query image by observing the index of all images with the same identity as the query image within the ranked gallery set, and concludes by calculating the mean. Hence, when the gallery set contains multiple images with the same identity as the probe image, using the mAP for evaluation will consider the ranked position of all gallery images with the same identity as the probe, rather than just the gallery image with the highest ranked position, as shown in Figure 2.12. For this reason, mAP is becoming more suitable when evaluating larger data sets with multiple images per identity.

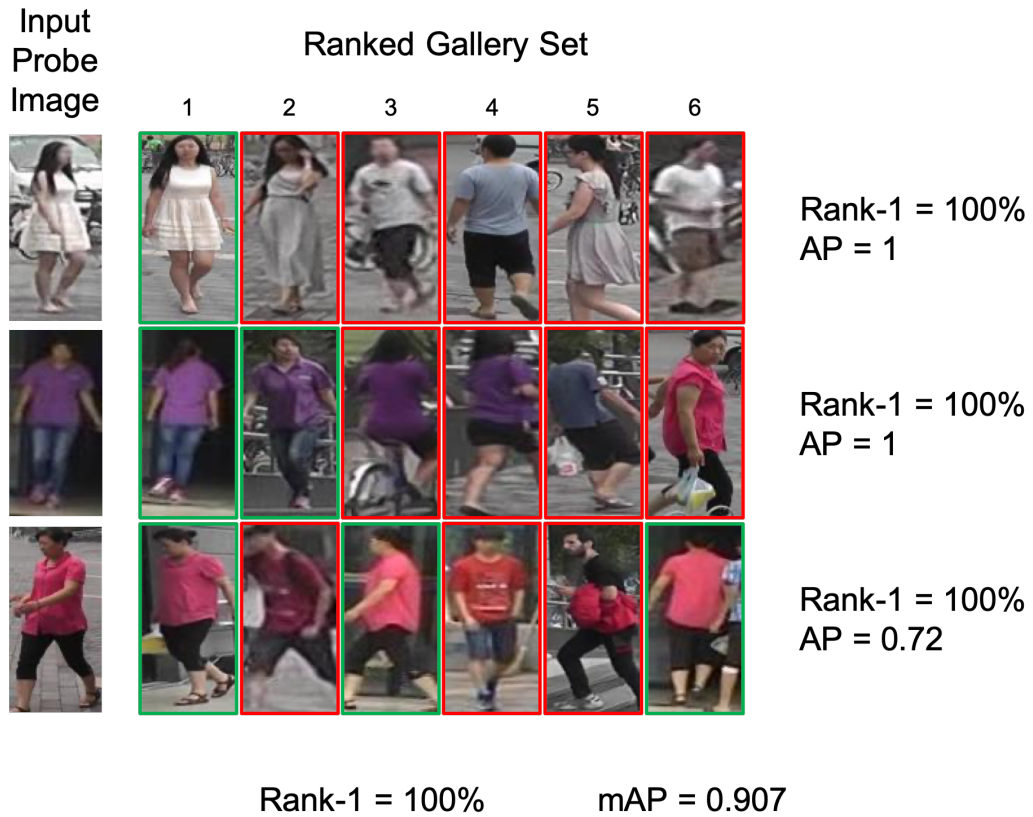


Figure 2.12: A comparison between rank- n and mAP/AP. It can be observed that whilst the rank-1 rate is consistently 100% throughout all examples, utilising mAP and AP better evaluates the performance when multiple images of the same identity as the input probe image are present within the gallery set. Images with the same identity as the input probe image are highlighted in green, whereas images with a different identity are highlighted in red. All images are from the Market-1501 [219] data set.

2.7 Data Sets

Images taken of even the same individual under different circumstances can vary significantly. Factors such as the type of camera used, the time of day and the weather at the time of an image being taken can contribute to significant visual differences present within an image. As such, a robust Re-ID system must be able to handle significant visual differences present within Re-ID images. A large number of data sets have been proposed which attempt to emulate real-world conditions and present a set of images with varying levels of illumination, pose, background, occlusion and blur. However, many data sets still suffer from being captured from a single location, and as such, variations in environment are limited. For this reason,

many Re-ID methods are trained using a combination of multiple data sets. Similarly, in order to prove their ability to generalise, many Re-ID methods are evaluated using multiple individual data sets.

Re-ID data sets consist of at least two images for each identity, generally cropped to the bounding box of the person. In order for a successful re-identification to occur, at least one image of the individual must belong to the probe set, and another to the gallery set. In addition, to increase the experimental difficulty, it is common for additional identities which are not present in the probe set to be included within the gallery set.

Furthermore, the use of standard data sets not only provides a means of evaluating individual Re-ID methods, but also for comparison between different methods. With the rise of deep learning techniques requiring a greater number of training samples per identity, there has been a recent influx of new, larger Re-ID data sets used alongside older, smaller data sets. The following section will describe the most commonly used Re-ID data sets in greater detail.

2.7.1 VIPeR (2007)

The VIPeR data set was proposed by Gray et al. [59]. It was gathered over several months, and as such, has a large variation in illumination caused by fluctuation in weather conditions. It contains 632 identities, with two images of each identity taken from separate cameras. There are significant variations in pose present, and some cases of significant blur. Occlusion is mostly absent. All images are captured in an academic setting, are cropped to the bounding box of the person, and are scaled to 128×48 pixels. Examples of images from this data set can be seen in Figure 2.13.



Figure 2.13: Example images from the VIPeR [59] data set. Each column represents a single identity. All images have been scaled to a standard resolution.

2.7.2 QMUL GRID (2009)

The QMUL GRID [117, 124, 125] data set contains 250 identities, with two images per identity, taken from separate cameras. In addition, there are 500 additional distinct identities with only a single image per identity, which is often used to expand the gallery set. All images are taken from an underground stations, and are cropped to the person. Images within this data set contain significant variations in pose, illumination and background, and contain significant amounts of blur and occlusion. Furthermore, there is some small variation in image resolution. Examples of images from this data set can be seen in Figure 2.14.



Figure 2.14: Example images from the QMUL GRID [117, 124, 125] data set. Each column represents a single identity. All images have been scaled to a standard resolution.

2.7.3 CUHK (2012-2014)

CUHK refers to three data sets named CUHK01 [110], CUHK02 [109] and CUHK03 [111]. The CUHK01 data set, otherwise known as the Campus data set, contains 971 identities with 2 images per identity. CUHK02 contains 1816 identities across five cameras, with each identity having two images per camera view. CUHK03 contains 1467 identities photographed from five pairs of camera views, with each identity therefore having up to ten images. The images within all three data sets are manually cropped to the bounding box of the person, however CUHK03 also offers a version of the data set automatically cropped by a person detector. Images within these data sets have significant variation in pose, illumination and background, as well as large amounts of occlusion and blur. Examples of images from this data set can be seen in Figure 2.15.



Figure 2.15: Example images from the CUHK01 [110] data set. Each column represents a single identity. All images have been scaled to a standard resolution.

2.7.4 PRID2011 (2011)

PRID2011 was proposed by Hirzer et al. [70], and contains both single-shot and multi-shot variations. The data set consists of two camera views, with the first containing 385 identities, and the second containing 749 identities. The first 200 identities are common to both camera views. Images from this data set have large variations in pose and illumination, yet do not possess significant blur or occlusion. All images are cropped to the individual, and are scaled to a resolution of 128×64 pixels. Examples of images from this data set can be seen in Figure 2.16.



Figure 2.16: Example images from the PRID2011 [70] data set. Each column represents a single identity. All images have been scaled to a standard resolution.

2.7.5 3DPeS (2011)

3DPeS [6–8] contains 193 identities taken from multiple cameras. It is a multi-shot data set, containing an unfixed number of images per identity. The data set contains a significant amount of pose, illumination and background variation. Images are

mostly cropped to the bounding box of the person, but image resolution is not consistent between images. Unlike most other Re-ID data sets, 3DPeS also offers binary masks to separate the foreground from the background. Examples of images from this data set can be seen in Figure 2.17.



Figure 2.17: Example images from the 3DPeS [6–8] data set. Each column represents a single identity. All images have been scaled to a standard resolution.

2.7.6 i-LIDS (2009) and iLIDS-VID (2014)

i-LIDS [57, 225, 226] consists of 476 images of 119 pedestrians filmed within an airport hall. iLIDS-VID [187] is created from the original source used for the i-LIDS data set, and consists of 300 identities with two image sequences per identity across two cameras. The images in both data sets contain significant variants in pose and background, as well as high amounts of occlusion. Illumination variation is a very big issue within this data set. Whilst the images are cropped to the person, the cropping often removes the outer edges of a person limbs. Examples of images from this data set can be seen in Figure 2.18.



Figure 2.18: Example images from the i-LIDS [57, 225, 226] data set. Each column represents a single identity. All images have been scaled to a standard resolution.

2.7.7 Market-1501 (2015)

Market-1501 [219] contains 32,668 images of 1,501 identities. It is captured from six cameras of varying quality, and possesses significant issues with pose and illumination variation. Also, the data set contains a lot of instances of occlusion, particularly by bicycles and bags. Many images are high quality, whereas many images also suffer from high levels of blur. A Deformable Part Model (DPM) [44] is used to crop pedestrian images to the bounding box of the person. Images are scaled to 128×64 pixels, and suffer from significant variations in pose, illumination and background. Also, oftentimes, the cropping process can remove parts of the persons body from the image. Examples of images from this data set can be seen in Figure 2.19.



Figure 2.19: Example images from the Market-1501 [219] data set. Each column represents a single identity. All images have been scaled to a standard resolution.

2.7.8 DukeMTMC-reID (2017) and DukeMTMC4reID (2017)

The DukeMTMC [152] data set is a video data set created using eight distinct cameras at the Duke University campus. DukeMTMC-reID [152, 228] was created using a subset of the images captured from DukeMTMC, cropping the images to the pedestrian’s bounding box. The data set consists of 36,411 bounding boxes, captured from 1,404 identities which appear in multiple cameras and 408 identities who appear in only one camera. Images from this data set possess similar visual characteristics to Market-1501, with images being cropped to 128×64 pixels, and possessing strong variations in pose, illumination and background. Examples of images from this data set can be seen in Figure 2.20.

A variation on DukeMTMC, named DukeMTMC4reID, was proposed in [56]. The authors use an off-the-shelf person detector [11] to locate the individual people, and produce 22,515 bounding boxes of 1,413 identities which appear in more than one camera, with 2,195 bounding boxes of 439 identities which appear in only one

camera. Also, there are 21,551 False Positive (FP) bounding boxes. As the source of the data set is communal with DukeMTMC-reID, DukeMTMC4reID shares the same visual characteristics with DukeMTMC-reID. However, the image resolution of the bounding boxes extracted from the person detector in DukeMTMC4reID varies from 72×34 to 415×188 pixels.



Figure 2.20: Example images from the DukeMTMC-reID [152, 228] data set. Each column represents a single identity. All images have been scaled to a standard resolution.

2.7.9 PETA (2014)

The PETA [39] data set consists of a combination of common Re-ID data sets, however, each identity is annotated with 61 binary and 4 multi-class attributes, such as the colour of a person’s clothing, or the length of their hair. As well as the aforementioned VIPeR, 3DPeS, CUHK, QMUL GRID, i-LIDS and PRID2011 data sets, it includes the following data sets:

- CAVIAR4REID [29] (2011) - CAVIAR4REID contains images of 72 people, with 50 appearing in multiple camera views, and the remaining 22 appearing in just one camera view. It suffers heavily from pose, illumination and background variation, and also with strong cases of blur. Images also vary significantly in resolution.
- MIT [142] (2000) - MIT consists of 888 identities with one image per identity. Image resolution is consistent. People present within this data set are generally either photographed from the front or the back with little variation in pose. Images are clear and well-illuminated, and occlusion is rare.
- SARC3D [8] (2011) - SARC3D consists of 50 identities with four images per identity. One image is taken from the front, back, left and right view of each identity. The visual characteristics of the images present within this data set

vary strongly between images, with pose and illumination varying the most significantly. In addition, image resolution is not consistent.

- TownCentre [12–14] (2009) - TownCentre consists of 6,967 images of 222 identities, taken from short video sequences. Image resolution varies significantly, as does the illumination and clarity of the images.

Examples of images from this data set can be seen in Figure 2.21.



Figure 2.21: Example images from the PETA [39] data set. All images have been scaled to a standard resolution. Individual images are originally part of [6–8, 12–14, 29, 57, 59, 70, 109–111, 117, 125, 142, 225, 226].

2.7.10 Limitations

Whilst images within individual data sets can contain significant variation in visual characteristics, some aspects such as environmental variation can be absent. For example, QMUL GRID contains images taken exclusively from within an underground transport station, whilst images from the CUHK data sets are taken exclusively from a university campus. Similarly, some data sets are not sufficiently large enough to be able to cover a significant amount of environmental variation. For these reasons, many methods [2, 62, 144, 206] combine multiple Re-ID data sets for training, ensuring significant environmental variation is present within their training set. Similarly, Re-ID methods are evaluated on multiple data sets to prove their generalisation capabilities.

Furthermore, Re-ID data sets mostly contain a small number of samples per person, and therefore temporal analysis of longer video sequences are not possible. Older Re-ID data sets can contain as little as one image per person per camera, whilst newer data sets often contain a significantly higher number of images per person per camera. As such, research into incorporating temporal data into the Re-ID process is now becoming more wide-spread.

2.8 Summary

In this chapter, we have reviewed various methods for modelling the foreground of a person image, motivated by the need to reduce the significant visual variations present within Re-ID data sets. Furthermore, for the task of building a robust feature descriptor to represent a person image, several feature extraction methods have been reviewed, including both hand-crafted and deep feature extraction methods. Other relevant works reviewed were distance metric learning and generative adversarial networks. Also, metrics for measuring the performance of Re-ID methods were introduced, as well as the standard data sets on which Re-ID methods are evaluated. The limitations of these data sets were also discussed. In the next chapter, our first major contribution is introduced, which is a method of foreground modelling using Partial Least Squares regression.

Chapter 3

Partial Least Squares Appearance Modelling

3.1 Introduction

Variations in visual characteristics of an image can significantly hinder the success of a Re-ID method. Therefore, a successful Re-ID method will contain some form of normalisation between images to counteract such variations. Pose and background variations can be reduced by a foreground model, learning which parts of a person image correspond to the person themselves rather than potentially redundant background information. Furthermore, modelling the foreground of a person image at a limb-by-limb level allows limbs of one person to be matched with the corresponding limbs of another.

However, Re-ID training data sets typically consists of person images with no corresponding foreground mask. Therefore, we label a set of twenty-nine keypoints on each training image to represent the skeleton and limb-widths of each person, and use this data to learn a mapping between image appearance features and skeleton keypoints, allowing a skeleton to be predicted by our model when presented with an unseen image.

In this chapter, we propose a Partial Least Squares Appearance Model [189] fitting approach which is combined with robust feature extraction and distance metric learning stages to match individuals. In Section 3.2, we describe building our skeleton prediction model using Partial Least Squares (PLS) Regression [127, 155] in order to learn which areas represent the person’s body. We then extract features and weight them to favour those extracted from foreground areas, explained in Section 3.4. Section 3.5 describes how we learn a metric for calculating the distance between feature descriptors. We validate our results against other methods in Section 3.6.

Finally, we summarise our findings in Section 3.7.

3.2 Partial Least Squares Foreground Appearance Modelling

To predict the skeleton of a person within a person image, we use Partial Least Squares (PLS) to learn a regression between the appearance of a person image and a set of labelled keypoints representing a person’s skeleton.

As a skeleton consists of multiple keypoints, this is a multi-target problem. Given a response matrix Y and a predictor matrix X , Multiple Linear Regression (MLR) aims to learn a linear relationship between these variables such that:

$$y_{ik} = \beta_{0k} + \sum_{j=1}^N \beta_{jk} x_{ij} + \epsilon_{ik}, \quad (3.1)$$

$$Y = X\beta + \epsilon$$

where β is a matrix consisting of regression coefficients, $i \in \{1, \dots, I\}$ where I is the number of samples, $k \in \{1, \dots, K\}$ where K is the number of response variables, and ϵ is an error term [67].

Regression methods such as MLR utilise an objective function such as Ordinary Least Squares (OLS), which minimises the sum of squares of the difference between the ground-truth and values predicted by the regression model:

$$\min \sum_{i=1}^I \sum_{k=1}^K (y_{ik} - \beta_{0k} - \sum_{j=1}^N \beta_{jk} x_{ij})^2. \quad (3.2)$$

However, this causes outliers to have a disproportionately large influence on the parameters of the OLS model, as the difference between an outlier’s ground-truth and predicted value are likely to be larger. Furthermore, OLS models have been shown to produce poor results when dealing with high-dimensional data and collinearity [61, 210]. Ridge Regression [178] utilises an extension of OLS which penalises large values of β , by adding a weighted ℓ_2 norm of β to the objective function. This is a form of regularization, and is often used to prevent overfitting by minimising the complexity of the model [158]. Unlike OLS, Ridge Regression can handle data with high levels of collinearity [41], but operates natively on the full input data without any dimensionality reduction. Principal Component Regression (PCR) [87] is an extension of MLR, and utilises Principal Component Analysis (PCA) [197] to reduce the dimensionality of the input data. However, methods such as PCR involve eigen-decomposition of the predictor matrix retaining the

principal components with the highest variance without paying consideration to the corresponding process with the response matrix, and vice-versa. Hence, the extracted principal components may not be optimal for predicting the response matrix. In addition, these methods suffer from poor results if presented with predictor matrices with collinearity [196].

PLS, however, is particularly useful for modelling predictor data with strongly collinearity, as well as in situations where the predictor matrix is high-dimensional [198]. Within the context of skeleton fitting for Re-ID, the predictor matrix consists of high-dimensional features extracted from image data, and is also highly collinear, given the relationship between neighbouring image regions. In addition, unlike PCA, PLS performs decomposition on both the predictor and response matrices simultaneously, considering how each of the predictor variables influences each of the target variables during the dimensionality reduction stage. Hence, PLS is more likely to extract principal components important for regression. For these reasons, we choose PLS to build our regression model, as we believe that it is the most suited to the skeleton prediction task. However, there are other methods, such as Canonical Correlation Analysis (CCA) [73], which also perform dimensionality reduction whilst considering the relationship between both the predictor and response matrices [175], and hence may also be suitable for application to this problem [89]. An alternative approach would be to utilise a neural network for the skeleton prediction task, such as a multilayer perceptron (MLP). In particular, Convolutional neural networks (CNNs) have been used extensively in recent years for computer vision tasks, showcasing promising results [2, 27, 28, 51, 108, 111, 112, 200, 203, 206, 229, 230]. Therefore, we will evaluate skeleton prediction using neural networks in Chapter 4.

As with Multiple Linear Regression (equation 3.1), PLS also builds a linear model $Y = X\beta + \epsilon$. However, PLS also solves:

$$\begin{aligned} X &= TP^T + E \\ Y &= UQ^T + F \end{aligned} \tag{3.3}$$

where T and U are the X and Y score matrices respectively, P and Q are the X and Y loading matrices respectively, and E and F are independent error terms. PLS can be solved by the SIMPLS [37] method, which begins by calculating X_0 and Y_0 by centering both X and Y , i.e. subtracting from them their mean values:

$$\begin{aligned} X_0 &= X - \text{mean}(X) \\ Y_0 &= Y - \text{mean}(Y) \end{aligned} \tag{3.4}$$

Next, the cross-correlation of the centred input matrices, Z_0 , is computed by:

$$Z_0 = X_0^* Y_0 \quad (3.5)$$

where X_0^* represents the complex conjugate transpose of matrix X_0 . Given an input number of components $ncomp$, we iterate through $a = (0, \dots, ncomp - 1)$, computing equation 3.6 to equation 3.9 for each iteration.

Firstly, we calculate and normalise the weight vector of the X matrix, \mathbf{r}_a , and the scores of the X matrix, \mathbf{t}_a , as follows:

$$\begin{aligned} \mathbf{r}_a &= Z_a \mathbf{e}_a \\ \mathbf{t}_a &= X_0 \mathbf{r}_a \\ \mathbf{t}_a &= \mathbf{t}_a - \text{mean}(\mathbf{t}_a) \\ \mathbf{t}_a &= \frac{\mathbf{t}_a}{\sqrt{\mathbf{t}_a^* \mathbf{t}_a}} \\ \mathbf{r}_a &= \frac{\mathbf{r}_a}{\sqrt{\mathbf{t}_a^* \mathbf{t}_a}} \end{aligned} \quad (3.6)$$

where \mathbf{e}_a is the most dominant eigenvector of $Z_a^* Z_a$, equivalent to the right singular vector of Z_a , with corresponding dominant singular vector equaling the maximum attainable covariance. The score matrix consists of PLS components which represent linear combinations of variables within X_0 , whilst the values with the weight matrix are used to calculate the scores, ensuring that the covariance of the two score matrices is maximised. We further calculate loadings of the X matrix, \mathbf{p} , the loadings of the Y matrix, \mathbf{q} , and the scores of the Y matrix, \mathbf{u} , as follows:

$$\begin{aligned} \mathbf{p}_a &= X_0^* \mathbf{t}_a \\ \mathbf{q}_a &= Y_0^* \mathbf{t}_a \\ \mathbf{u}_a &= \begin{cases} Y_0 \mathbf{q}_a - T(T^*(Y_0 \mathbf{q}_a)), & \text{if } a > 0 \\ Y_0 \mathbf{q}_a, & \text{otherwise} \end{cases} \end{aligned} \quad (3.7)$$

The loadings of the X matrix and the loadings of the Y matrix are calculated such that $\mathbf{t}_a \mathbf{p}_a^*$ and $\mathbf{t}_a \mathbf{q}_a^*$ are the PLS approximations to X_0 and Y_0 respectively.

Additionally, a variable v is defined and normalised to allow for the aforementioned cross-correlation of the original input matrices, Z , to be further deflated, by:

$$\mathbf{v}_a = \begin{cases} \mathbf{p}_a - V(V^*\mathbf{p}_a), & \text{if } a > 0 \\ \mathbf{p}_a, & \text{otherwise} \end{cases} \quad (3.8)$$

$$\mathbf{v}_a = \frac{\mathbf{v}_a}{\sqrt{\mathbf{v}_a^*\mathbf{v}_a}}$$

The deflation of matrix Z allows for the successive most dominant eigenvectors to be computed through eigenanalysis. The cross-correlation matrix Z is therefore deflated as follows:

$$Z_{a+1} = Z_a - \mathbf{v}_a(\mathbf{v}_a^*Z_a) \quad (3.9)$$

Following the process from equation 3.6 to equation 3.9 being repeated for values of \mathbf{a} from 0 to $\text{ncomp}-1$, the PLS coefficients β are computed as follows:

$$\begin{aligned} \beta &= RQ^* \\ \mathbf{k} &= \text{mean}(Y) - \text{mean}(X)\beta \\ \beta &= (\mathbf{k}; \beta) \end{aligned} \quad (3.10)$$

where \mathbf{k} is a row vector concatenated to β as the first row, to allow calculation of the intercept terms, $\beta_{0,*}$. The SIMPLS algorithm [37] works to maximise the covariance between the scores of the Y matrix, \mathbf{u}_a , and the scores of the X matrix, \mathbf{t}_a , such that:

$$\underset{\mathbf{r}_a, \mathbf{q}_a}{\text{argmax}} \text{cov}(\mathbf{u}'_a, \mathbf{t}_a) = \underset{\mathbf{r}_a, \mathbf{q}_a}{\text{argmax}} \text{cov}(\mathbf{q}_a^*(Y_0^*X_0)\mathbf{r}_a) \quad (3.11)$$

subject to $\mathbf{r}_a^*\mathbf{r}_a = 1$, $\mathbf{q}_a^*\mathbf{q}_a = 1$, and $\mathbf{t}_b^*\mathbf{t}_a = 0$ for $a > b$. The SIMPLS method is summarised in Algorithm 1.

When presented with a variable containing the appearance features of an unseen person, X_i , the learnt PLS regression model can predict the person's skeleton, \hat{Y}_i , by evaluating:

$$\hat{Y}_i = (1, X_i)\beta \quad (3.12)$$

where a value of 1 is concatenated to X_i in order to compute the intercept terms, $\beta_{0,*}$.

In our experiments, the appearance information X is represented by His-

Input : A response matrix X , a predictor matrix Y , and a number of components $ncomp$.

Output : PLS regression coefficients β .

Initialise: R , T , P , Q , U and V .

$$X_0 = X - \text{mean}(X)$$

$$Y_0 = Y - \text{mean}(Y)$$

$$Z_a = X_0^* Y_0$$

Calculate the cross-correlation of the input matrices.

for $a = 0$; $a < ncomp$; $a = a + 1$ **do**

e_a = most dominant eigenvector of Z_a

$$r_a = Z_a e_a$$

Calculate X matrix weights.

$$t_a = X_0 r_a$$

Calculate X matrix scores.

$$t_a = t_a - \text{mean}(t_a)$$

$$t_a = \frac{t_a}{\sqrt{t_a^* t_a}}$$

$$r_a = \frac{r_a}{\sqrt{t_a^* t_a}}$$

Normalise X matrix weights and scores.

$$p_a = X_0^* t_a$$

Calculate X matrix loadings.

$$q_a = Y_0^* t_a$$

Calculate Y matrix loadings.

$$u_a = \begin{cases} Y_0 q_a - T(T^*(Y_0 q_a)), & \text{if } a > 0 \\ Y_0 q_a, & \text{otherwise} \end{cases}$$

Calculate Y scores, ensuring orthogonality.

$$v_a = \begin{cases} p_a - V(V^* p_a), & \text{if } a > 0 \\ p_a, & \text{otherwise} \end{cases}$$

$$v_a = \frac{v_a}{\sqrt{v_a^* v_a}}$$

$$Z_{a+1} = Z_a - v_a(v_a^* Z_a)$$

Deflate the cross-correlation of the input matrix.

$$R(:,a) = r_a$$

$$T(:,a) = t_a$$

$$P(:,a) = p_a$$

$$Q(:,a) = q_a$$

$$U(:,a) = u_a$$

$$V(:,a) = v_a$$

Store $r_a, t_a, p_a, q_a, u_a, v_a$ into R, T, P, Q, U and V respectively.

end

$$\beta = RQ^*$$

$$k = \text{mean}(Y) - \text{mean}(X)\beta$$

$$\beta = (k; \beta)$$

Compute the regression coefficients.

Algorithm 1: The SIMPLS [37] method.

togram of Oriented Gradients (HOG) features extracted on a regular grid across our training set, whilst Y is the output skeleton vector. We continue by learning the regression between the appearance information and the corresponding labelled skeleton keypoints. Once the skeleton has been predicted, it is used to create an image mask. If this image mask is applied to the original image, the foreground can be separated from the background. Figure 3.1 shows examples of the skeleton fitting using our PLS skeleton prediction model.

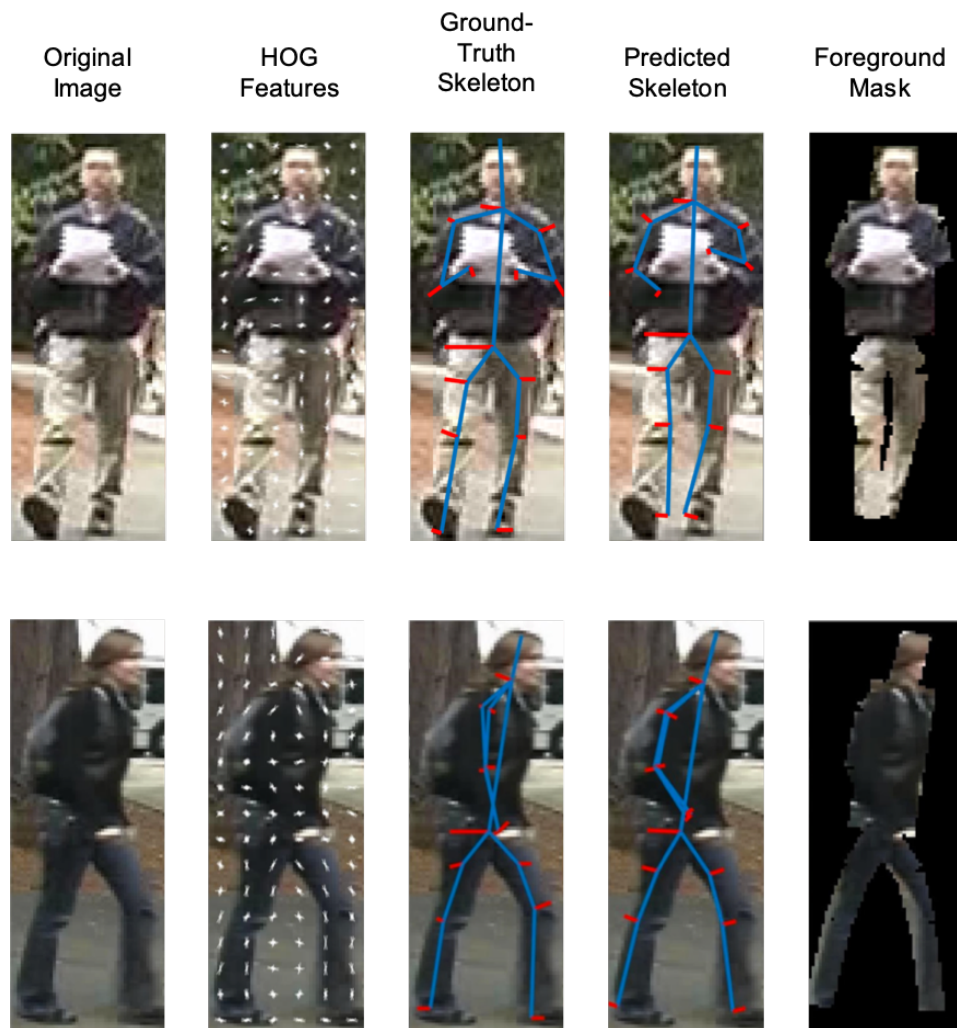


Figure 3.1: Examples of our PLS skeleton fitting on images from the VIPeR [59] data set. Each set of five images contains: the original image, the input HOG appearance features, the ground-truth skeleton keypoints, our predicted skeleton using the PLS regression model and the foreground image mask.

3.3 Partial Least Squares Orientation Modelling

There are a significant number of different poses represented within Re-ID data sets. Therefore, any appearance model must capture as many of these poses as possible. One of the greatest causes of pose variation within Re-ID data sets is the difference in person orientation relative to the camera. Therefore, in order to increase the accuracy of our skeleton fitting results, we divide each data set into n distinct orientation groups based on ground-truth orientation labelling. Figure 3.2 shows examples of the different orientation groups within the VIPeR data set, which we divided into two groups: the first consists of those facing perpendicular (90°) to the camera, whilst the second consists of those facing all other directions relative to the camera.



Figure 3.2: Examples of orientation groups from the VIPeR [59] data set. For this data set, we split the images into two orientation groups: those facing perpendicular (90°) to the camera, and those facing all other directions relative to the camera.

The use of distinct orientation groups allow us to create a separate PLS appearance model for each group. As such, each PLS appearance model can be trained on and specialise in predicting the skeleton of one particular orientation group, increasing overall skeleton prediction results. Figure 3.3 shows examples of skeletons predicted using various PLS appearance models.

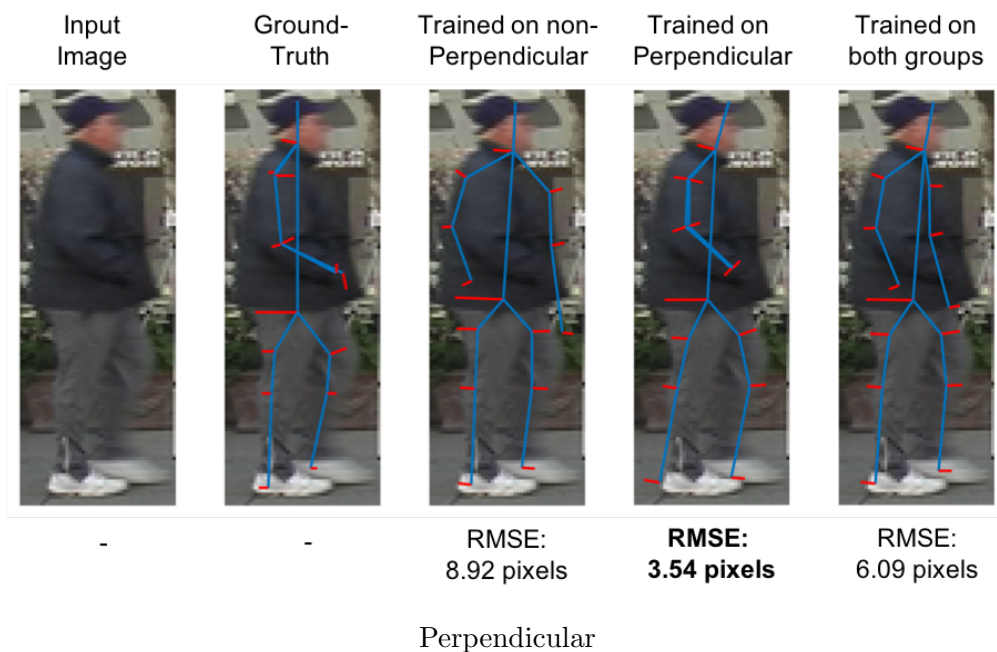
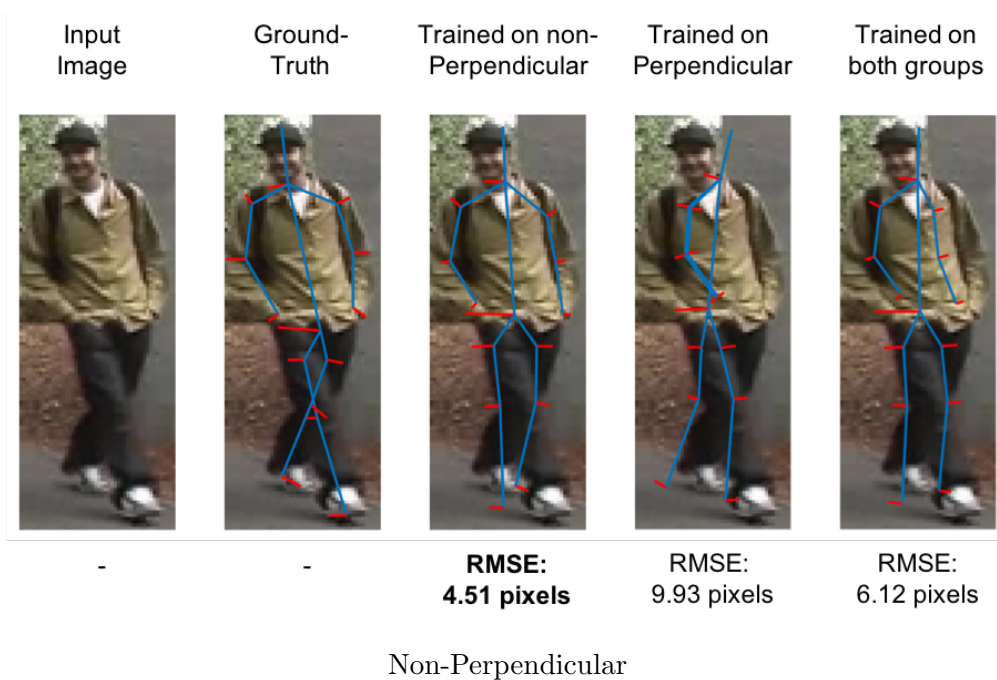


Figure 3.3: Examples of training the PLS appearance model on different subsets of the VIPeR [59] data set. The third column shows training only on images containing people with non-perpendicular orientations relative to the camera. The fourth column represents training on images containing only images containing people with perpendicular orientations relative to the camera. The fifth and final column shows training on images containing people with all poses relative to the camera. It can be seen that the lowest RMSE is obtained by training on images with similar orientations to the input image.

As can be seen in Figure 3.3, applying the PLS appearance model representing a different orientation to the one present within the input image can lead to poor fitting results, and by extension, poor foreground modelling. This may therefore lead to “foreground” feature descriptors being extracted from non-foreground areas, lowering rank- n matching rates. Thus, we aim to predict the skeleton of an image with a model trained on images with a similar orientation, and hence achieve the most accurate skeleton fitting.

However, given an unseen test image, the use of multiple PLS appearance models mandates the need to classify an image into one of n orientation groups. We therefore utilise mixture of experts [82] and create an additional model for the purpose of predicting the orientation group of an input image. For this task, we choose to again use a PLS-based model according to equation 3.1, where X is equal to the HOG appearance features as extracted in Section 3.2, and $Y \in \{0, \dots, n - 1\}$, where n is the number of distinct orientation groups. We choose to again utilise PLS in order to evaluate its performance as a means of regressing between appearance features and an appropriate orientation group, however, an alternative would be to use a classification model such as SVM [31] or Naïve Bayes [128, 134]. Following the classification of an input image into an orientation group, the PLS skeleton prediction model corresponding to that orientation group is used to predict the skeleton of the input image.

3.4 Feature Extraction and Weighting

In this section, we will explain how we combine foreground modelling and feature extraction to improve matching rates. We first extract the Local Maximal Occurrence (LOMO) [115] feature descriptors as our baseline, and then concatenate with our proposed features to form the Weighted LOMO (Section 3.4.3) feature descriptor. Additionally, we extract Salient Colour Names Based Colour Descriptor (SCNCD) [208] feature descriptors on a limb-by-limb basis, and concatenate to form a foreground feature descriptor (Section 3.4.2). We concatenate the Weighted LOMO feature descriptors with the limb-by-limb level SCNCD feature descriptors to form our final feature descriptor for each image.

3.4.1 LOMO

Originally proposed by Liao et al. [115], the LOMO feature descriptor is a hand-crafted feature, consisting of both colour and textural information.

The Retinex Transform

In order to minimise the effects of illumination variation, the images are first passed through a Retinex transform [84, 85, 98, 133, 145] as a preprocessing step. Retinex processes an image and recreates it in a manner which is consistent with human observation of the scene, restoring the image in a way which often leads to enhanced details in darker and shadowed areas.

The output of a Single-Scale Retinex, $R_i(x, y)$, is given by:

$$R_i(x, y) = \log I_i(x, y) - \log[F(x, y) * I_i(x, y)] \quad (3.13)$$

where $I_i(x, y)$ is the image distribution in the i th spectral band, (x, y) are the pixel coordinates, $*$ is the convolution operation, and $F(x, y)$ is a low-pass filter denoted as the surround function:

$$F(x, y) = K e^{-\frac{(x^2+y^2)}{\sigma^2}} \quad (3.14)$$

where σ is the Gaussian surround space constant, and K is chosen such that:

$$\iint F(x, y) dx dy = 1 \quad (3.15)$$

However, Single-Scale Retinex transformations are unable to simultaneously achieve strong dynamic range compression and colour rendition, causing images to often appear with detail hidden in darker and shadowed areas, or to have poor colour reproduction. Low values of σ achieve better dynamic range compression, whilst high values achieve better colour rendition. A solution to this problem is to apply a Multi-Scale Retinex, which is a weighted sum of Single-Scale Retinex outputs, utilising various different values of σ , such that:

$$R_{MSR_i} = \sum_{j=1}^J w_j R_{j_i} \quad (3.16)$$

where J is the number of scales, and w_j is the weight associated with the corresponding scale. R_{j_i} is the i^{th} component of the j^{th} scale. For the Multi-Scale Retinex algorithm, the surround function is instead defined as:

$$F_n(x, y) = K e^{-\frac{(x^2+y^2)}{\sigma_n^2}} \quad (3.17)$$

However, the above preprocessing has the negative effect of significantly “greying-out” the image. Given Re-IDs reliance on colour information, such severe desaturation of images would be largely detrimental to the Re-ID process. Therefore,

the Multi-Scale Retinex process also contains a colour restoration method:

$$R_{MSRCR_i} = C_i(x, y)R_{MSR_i} \quad (3.18)$$

where

$$\begin{aligned} I'_i(x, y) &= \frac{I_i(x, y)}{\sum_{i=1}^S I_i(x, y)} \\ C_i(x, y) &= f[I'_i(x, y)] \end{aligned} \quad (3.19)$$

i represents a specific colour band, S is the number of spectral channels, and $C_i(x, y)$ is the i th band of the Colour Restoration Function (CRF), which is defined as:

$$C_i(x, y) = \beta \log[\alpha I'_i(x, y)] \quad (3.20)$$

where β is a gain constant, whilst α controls the strength of the nonlinearity. However, the output of this equation is in the logarithmic domain, hence gain and offset variables are required. Thus, the final equation for computing the Multi-Scale Retinex with Colour Restoration is:

$$R_{MSRCR_i} = G[C_i(x, y)R_{MSR_i} + b] \quad (3.21)$$

where G and b are the gain and offset variables respectively. Figure 3.4 shows examples of Re-ID images before and after the Retinex preprocessing step has been applied. All images shown in Figure 3.4 demonstrate greater detail, specifically in darker areas such as torso areas. In (a), greater detail is visible in the person's face and torso. In (b), it can be seen that further detail is extracted from the persons hair, whilst in (c) the red detail is emphasised in the persons shorts. The white patch and coat is emphasised in (d), whilst the texture on the persons shirt is made more pronounced in (e). In (f), greater detail can be seen in the persons bag strap and torso.



Figure 3.4: Example image pairs from the VIPeR [59] data set, containing both the original images as well as the images after the Retinex preprocessing step has been applied.

LOMO features apply the Multi-Scale Retinex transform, with two scales of a centre/surround Retinex where $\sigma = 5$ and $\sigma = 20$.

Feature Extraction & Managing Viewpoint Variation

LOMO employs both colour and textural information to construct a feature descriptor from the preprocessed images. Images represented using the RGB model mix both colour and luminosity information, and therefore are not optimal for use in feature extraction. For this reason, images are converted to the HSV colour model prior to feature extraction. The SILTP [114] feature type is used for extracting textural information. SILTP is an improved version of Local Binary Patterns [138], extracting textural information from images whilst maintaining invariance to changes in intensity, as well as to image noise. Given an image location (x_c, y_c) , the SILTP feature is calculated as:

$$SILTP_{N,R}^{\tau}(x_c, y_c) = \bigoplus_{k=0}^{N-1} s_{\tau}(I_c, I_k), \quad (3.22)$$

where I_c is the gray intensity value of the centre pixel, I_k are the gray intensities of its N neighbourhood pixels located equally spaced on a circle of radius R , \bigoplus is the concatenation operation of binary strings, τ is a scale factor, and s_{τ} is the function:

$$s_{\tau}(I_c, I_k) = \begin{cases} 01, & \text{if } I_k > (1 + \tau)I_c \\ 10, & \text{if } I_k < (1 + \tau)I_c \\ 00, & \text{otherwise} \end{cases} \quad (3.23)$$

To maintain spatial information within the feature descriptor, a sliding window approach is used, with images divided into 10×10 pixel windows with a 5 pixel overlap in the height and width dimension. From within each window, an $8 \times 8 \times 8$ joint HSV histogram is extracted, as well as two scales of SILTP [114] histograms, $SILTP_{4,3}^{0.3}$ and $SILTP_{4,5}^{0.3}$.

However, given the significant pose and viewpoint variation present within ReID images, corresponding windows between images may not represent the same parts of people's bodies. Thus, to minimise the effects of pose and viewpoint variations, LOMO takes all windows with the same horizontal location, and for each histogram bin, takes the maximum value across all windows to form the feature descriptor for that horizontal location.

To take into account multi-scale information, the images are downscaled by a factor of two and four, and the feature extraction process is repeated. Furthermore, a log transform is applied, and both HSV and SILTP features are normalised to a unit length. Figure 3.5 demonstrates the LOMO feature extraction process.

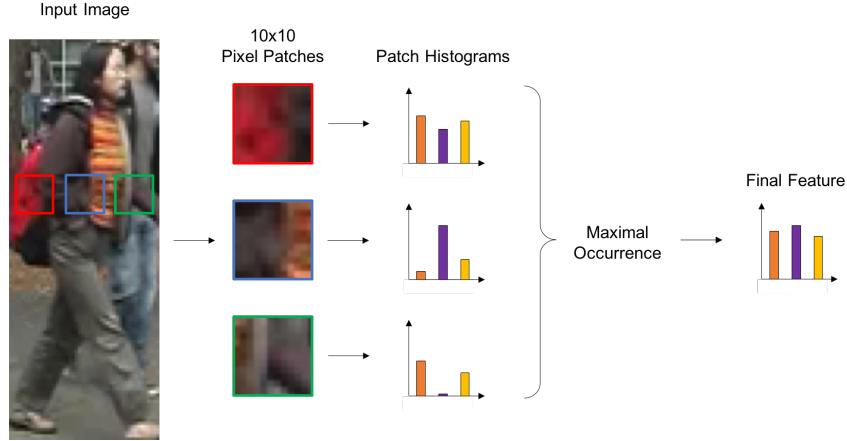


Figure 3.5: The LOMO [115] feature extraction method. The diagram is based on a similar diagram contained in [115].

3.4.2 Salient Colour Names Based Colour Descriptor (SCNCD)

Originally proposed by Yang et al. [208], the Salient Colour Names Based Colour Descriptor (SCNCD) defines sixteen colour co-ordinates in the RGB colour space. The colours are carefully chosen to contain a spread from across the RGB spectrum, including fuchsia, blue, aqua, lime, yellow, red, purple, navy, teal, green, olive, maroon, black, gray, silver and white. This is an extension of the Colour Names [181] method, which contained only eleven colours, including black, blue, brown, grey, green, orange, pink, purple, red, white, and yellow.

The first step is to quantise the RGB colour space into $32 \times 32 \times 32$ indexes, \mathbf{d} , with each index possessing 512 similar colours, \mathbf{w} . Thus, \mathbf{d} can be defined as $\mathbf{d} = \{\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_{512}\}$, which are the set of colours for which the index is the same. The set of colour names are defined as co-ordinates in the RGB colour space, $\mathbf{Z} = \{\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_{16}\}$, and a mapping/ posterior probability from an index \mathbf{d} and a colour distribution \mathbf{d} is generated. Similar to Bag of Words [64, 168], this process is a form of vector quantisation, and allows for multiple similar colours to be assigned to similar colour name distributions.

The probability of assigning \mathbf{d} to a colour name \mathbf{z} is:

$$p(\mathbf{z}|\mathbf{d}) = \sum_{n=1}^{512} p(\mathbf{z}|\mathbf{w}_n)p(\mathbf{w}_n|\mathbf{d}) \quad (3.24)$$

where $p(\mathbf{z}|\mathbf{w}_n)$ is a distribution of probabilities calculated as normally distributed variates of the closest K colour names from a given colour \mathbf{w}_n , where:

$$p(\mathbf{z}|\mathbf{w}_n) = \begin{cases} \frac{\exp(-\|\mathbf{z}-\mathbf{w}_n\|^2/\frac{1}{K-1}\sum_{\mathbf{z}_l \neq \mathbf{z}}\|\mathbf{z}_l-\mathbf{w}_n\|^2)}{\sum_{p=1}^K \exp(-\|\mathbf{z}_p-\mathbf{w}_n\|^2/\frac{1}{K-1}\sum_{\mathbf{z}_s \neq \mathbf{z}_p}\|\mathbf{z}_s-\mathbf{w}_n\|^2)}, & \text{if } z \in KNN(\mathbf{w}_n) \\ 0, & \text{otherwise} \end{cases} \quad (3.25)$$

The $p(\mathbf{w}|\mathbf{d})$ term in equation 3.24 weighs the contribution of \mathbf{w}_n to \mathbf{d} , ensuring that values of \mathbf{w}_n closer to the mean of \mathbf{w}_n , μ , contribute greater to \mathbf{d} .

$$p(\mathbf{w}_n|\mathbf{d}) = \frac{\exp(-\alpha\|\mathbf{w}_n - \mu\|^2)}{\sum_{l=1}^{512} \exp(-\alpha\|\mathbf{w}_l - \mu\|^2)} \quad (3.26)$$

The terms present in equation 3.24 capture the similarity between colour names and colour indexes \mathbf{d} . As similar colours result in similar colour name distributions, the effects of illumination variation can be minimised. We extract SCNCD features from all limbs of a given person separately, and concatenate to create a foreground descriptor. Figure 3.6 shows an example of SCNCD features being extracted from each limb separately. Similar to [208], we extract SCNCD features from the RGB, normalized RGB, *l1l2l3* [52] and HSV colour spaces. For each colour space, we then apply a log transform to the extracted features, and normalise to a unit length, before fusing to form our final limb-by-limb level SCNCD feature.

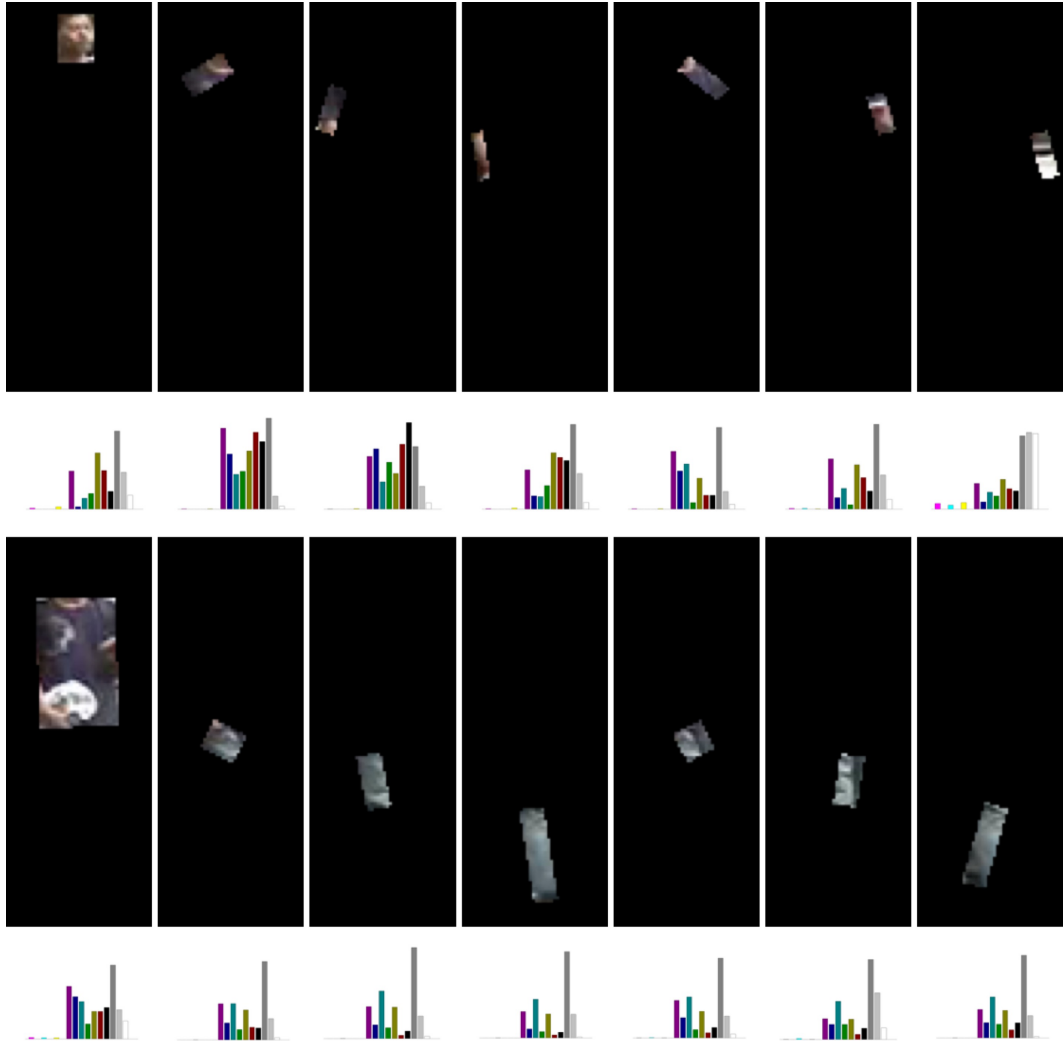


Figure 3.6: SCNCD [208] feature extraction at a limb-by-limb level. The histograms below each individual limb image represent the extracted feature descriptor, with each bar representing an individual colour name.

3.4.3 Weighted LOMO

Although the original LOMO features attempt to minimise the negative effects of pose and viewpoint variation, they cannot differentiate between different parts of a person’s body, such as on a limb-by-limb level, nor between foreground and background areas. We therefore aim to minimise the effects of pose and background variation by weighting LOMO features by incorporating the skeleton model described in Section 3.2.

As the predicted skeleton is likely to be different to a corresponding ground-truth skeleton, the probability of some foreground being considered background and

vice-versa is high. Therefore, we decide not to completely mask-out information considered background, due to the risk of inadvertently removing foreground information. Instead, we use the computed image mask to weight the LOMO features by the percentage of predicted foreground in each feature patch, by:

$$\mathbf{f}_w(B) = \frac{|F \cap B|}{|B|} \mathbf{f}(B) \quad (3.27)$$

where B is the set of pixels in the image patch and F is the set of pixels labelled foreground within the same image patch. Once all patches have been weighted, the maximum value for each histogram bin across each horizontal location is taken towards the final feature descriptor. As LOMO is a patch-based feature type, it was possible to weight each patch according to the percentage of foreground within the patch, allowing for the negative effects of poor skeleton prediction to be mitigated. This is in contrast to SCNCD, which is not patch-based, and requires a bounded image region from which to extract feature descriptors. To generate the LOMO features, we use the code given by [115], and alter it as described in Equation 3.27 to prioritise features from the foreground areas. We concatenate the original LOMO features with our new, weighted features to create Weighted LOMO. Figure 3.7 illustrates the weighting process. The concatenation of the Weighted LOMO and limb-by-limb level SCNCD feature descriptors form the final feature descriptor for each image.

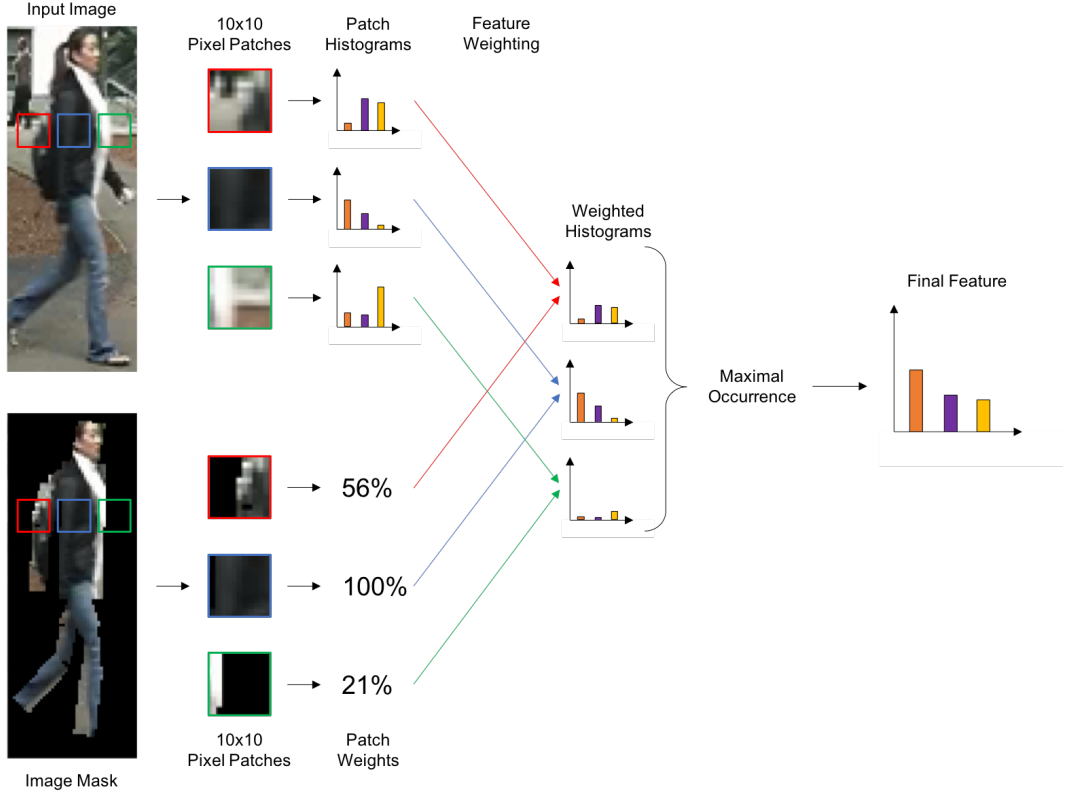


Figure 3.7: Our proposed method for weighting the LOMO feature descriptor. Patch features are extracted from each 10×10 pixel region, and then weighted by multiplying by the percentage of foreground pixels. Then, we take all patches in the same horizontal location and use the maximum value of each bin to contribute towards the final feature descriptor for that horizontal location.

3.5 Distance Metric Learning

Due to the significant intra-class variation visible within typical Re-ID imagery, traditional distance measures such as Euclidean are often inappropriate to use in a Re-ID context. As such, many Re-ID methods instead learn a custom distance metric optimal for the Re-ID task. Such distance metrics aim to bring feature descriptors of the same class closer to each other within the feature space.

KISSME [93] calculates the distance between two feature descriptors as:

$$\tau_M^2(\mathbf{f}_i, \mathbf{f}_j) = (\mathbf{f}_i - \mathbf{f}_j)^T M (\mathbf{f}_i - \mathbf{f}_j) \quad (3.28)$$

where $M = (\Sigma_I^{-1} - \Sigma_E^{-1})$ and Σ_I and Σ_E are the intra-personal and extra-personal scatter matrices respectively. Often, dimensionality reduction using techniques such as Principal Component Analysis (PCA) [197] is performed on the input

vectors prior to estimating Σ_I and Σ_E . However, these approaches do not consider the distance metric learning stage during the dimensionality reduction stage, and therefore are not optimal.

To increase optimality, we use Cross-view Quadratic Discriminant Analysis (XQDA) [115], a distance metric learning technique which extends KISSME, to perform our distance metric learning. Let D be the original dimensionality of the input data, and R being the reduced dimensionality. XQDA learns a subspace $W = (\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_r) \in \mathbb{R}^{D \times R}$, whilst simultaneously learning a distance function:

$$d_W(\mathbf{f}_i, \mathbf{f}_j) = (\mathbf{f}_i - \mathbf{f}_j)^T W (\Sigma_I'^{-1} - \Sigma_E'^{-1}) W^T (\mathbf{f}_i - \mathbf{f}_j) \quad (3.29)$$

where $\Sigma_I' = W^T \Sigma_I W$ and $\Sigma_E' = W^T \Sigma_E W$. Directly optimising d_W is difficult due to the presence of two inverse matrices. Furthermore, due to the distribution of intra-personal and extra-personal distances having zero mean, W cannot be determined using a traditional LDA approach. Instead, it is possible to maximise the variances σ_E and σ_I , to better distinguish between the two classes. The projection direction, \mathbf{w} , can be optimised such that $\sigma_E(\mathbf{w})/\sigma_I(\mathbf{w})$ is maximised. Since $\sigma_I(\mathbf{w}) = \mathbf{w}^T \Sigma_I \mathbf{w}$ and $\sigma_E(\mathbf{w}) = \mathbf{w}^T \Sigma_E \mathbf{w}$, the objective function is the Generalised Rayleigh Quotient:

$$J(\mathbf{w}) = \frac{\mathbf{w}^T \Sigma_E \mathbf{w}}{\mathbf{w}^T \Sigma_I \mathbf{w}}. \quad (3.30)$$

We can solve for \mathbf{w} by a generalised eigenvalue decomposition by maximising:

$$\max_{\mathbf{w}} \mathbf{w}^T \Sigma_E \mathbf{w}, \text{ s.t. } \mathbf{w}^T \Sigma_I \mathbf{w} = 1. \quad (3.31)$$

The largest eigenvalue of $\Sigma_I^{-1} \Sigma_E$ is equivalent to the maximum value of $J(\mathbf{w})$, with the corresponding eigenvector being the solution. The columns of W correspond to the R eigenvectors of $\Sigma_I^{-1} \Sigma_E$ taken in decreasing order of eigenvalue. Thus, with the above algorithm, XQDA learns a discriminant subspace, as well as a distance function within the learnt subspace.

3.6 Results and Discussion

In this section, we describe both our experimental settings and the Re-ID matching results obtained whilst using our proposed method. We utilise our hand-labelled skeleton keypoints (Appendix A), which consist of twenty-nine keypoints representing fourteen limbs, with each limb consisting of two end-points, as well as a third keypoint to represent the edge of the limb. The bottom keypoint of each limb also acts as the top keypoint of the following limb. When constructing our PLS regression models,

	PLSAM(v1) (Weighted LOMO only)	PLSAM(v2) (Weighted LOMO + Limb-by-Limb Level SCNCD)	LOMO only	Limb-by-Limb Level SCNCD only
VIPeR [59], QMUL GRID [117, 124, 125]	53,920	54,816	26,960	896
CUHK03 [111]	71,444	72,340	35,722	896

Table 3.1: The dimensionality of each of the feature type used in our experimentation. Due to the higher resolution available for most images within the CUHK03 [111] data set, we resize to 160×60 pixels for LOMO and Weighted LOMO feature extraction, rather than 128×48 pixels, resulting in a higher dimensional feature descriptor.

we choose the top fifteen components for the skeleton prediction models, and the top fifty components for the orientation prediction models.

We experiment on the following three data sets: VIPeR [59], QMUL GRID [117, 124, 125], and CUHK03 [111]. We extract Histogram of Oriented Gradients (HOG) [35, 47] features from Re-ID images prior to applying the Retinex (Section 3.4.1) preprocessing step, using a cell size of six pixels, and a block size of two pixels.

For the XQDA distance metric, we utilise the default hyperparameters as determined by [115]. The first hyperparameter, λ , ensures that Σ_I is not a singular matrix by adding λ to all elements of Σ_I , as otherwise this would result in Σ_I^{-1} being impossible to compute, and is set to 0.001 in our experiments. The second hyperparameter, `qdaDims`, is set to -1, which results in the subspace $W = (\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_r)$ consisting of the r eigenvectors of $\Sigma_I^{-1}\Sigma_E$ with corresponding eigenvalues greater than 1.

We define two feature descriptors to be used in our evaluation: PLSAM(v1) consists solely of the Weighted LOMO feature descriptors, whereas PLSAM(v2) consists of both the Weighted LOMO and limb-by-limb level SCNCD feature descriptors. We present the dimensionality of these feature descriptors alongside the dimensionality of the LOMO and limb-by-limb level SCNCD feature descriptors in Table 3.1.

3.6.1 Evaluation on the VIPeR data set

In this section, we present results of our method on the VIPeR [59] data set. We convert all images to the HSV colour space, and extract HOG features from the V channel to act as input for our PLS-based models. All images within the VIPeR data set are originally provided with a resolution of 128×48 pixels, and are not resized for our experiments. We train two separate skeleton prediction models, one

of which handles people facing perpendicular to the camera, and the other which handles those with alternative orientations. Furthermore, we train an orientation model to predict the orientation of a person, which therefore allows our method to choose which skeleton prediction model is most appropriate to use. We achieve this by utilising the orientation information provided by the VIPeR data set, which lists the angle each person is facing relative to the camera. By dividing the VIPeR data set into two equal sized sets, with 316 training identities and 316 testing identities, our orientation model achieves an accuracy of 91.9%.

Figure 3.8 shows examples of the best and worst skeleton fitting results on the VIPeR data set. We can see from this figure that the skeleton prediction model works well for more standard and common poses, but struggles when it comes to unusual poses such as a person raising their arms above their head. Figure 3.9 shows the distribution of RMSE on skeletons predicted using the PLS-based skeleton prediction model. Figure 3.10 shows a sample of images where our PLS-based method achieves good skeleton fitting results, whilst Figure 3.11 shows a sample of images where our PLS-based method achieves poor skeleton fitting results.

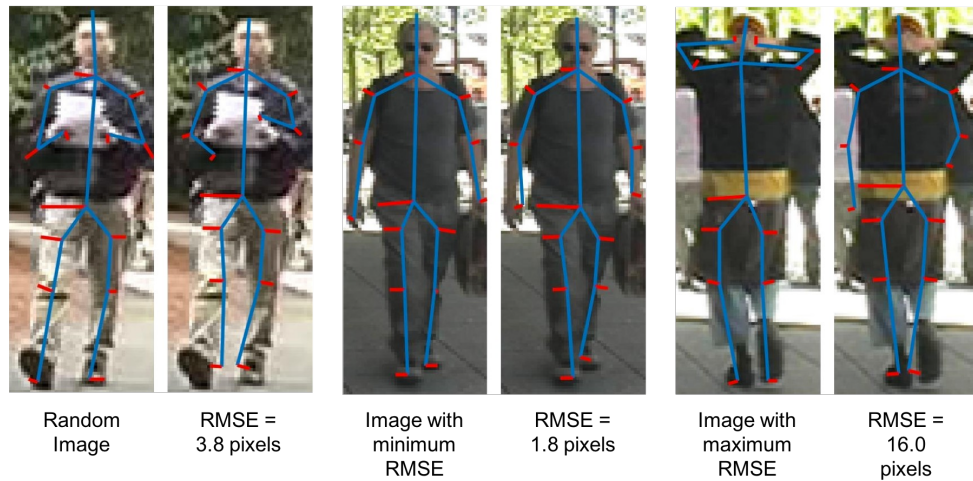


Figure 3.8: Examples of ground-truth and predicted skeletons from the VIPeR [59] data set. The average RMSE over the entire test set is 5.2 pixels.

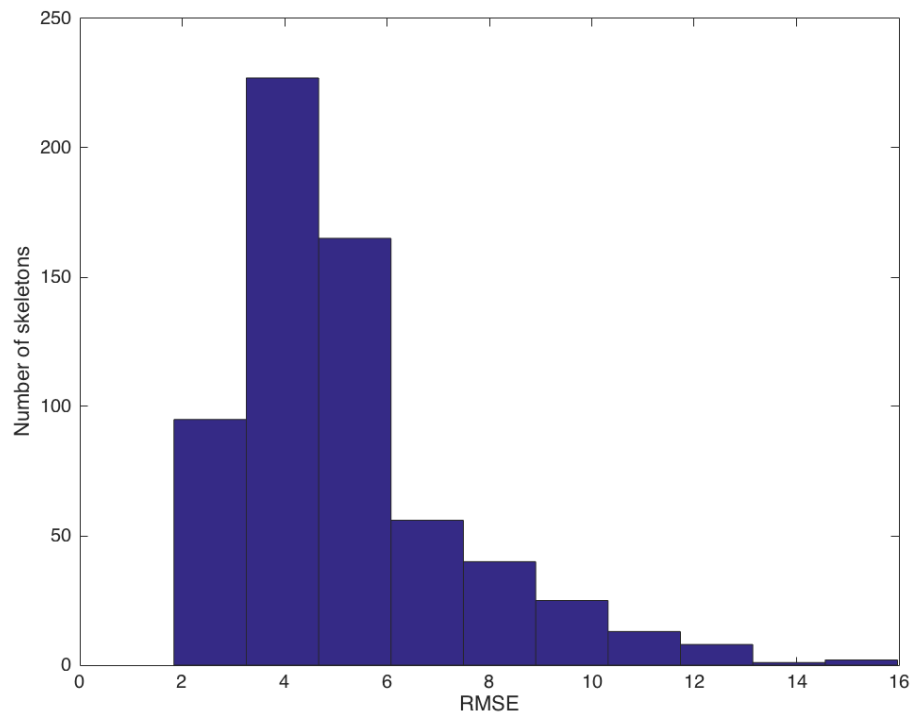


Figure 3.9: The distribution of RMSE on skeletons predicted by the PLS-based skeleton prediction model on the VIPeR [59] data set.








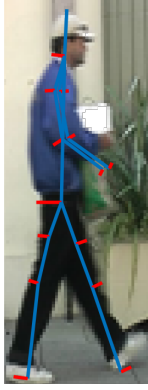


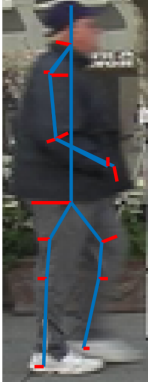
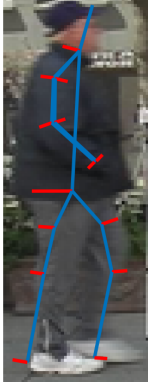
Good Fitting Results (VIPeR)			
Ground-Truth	PLS	Ground-Truth	PLS
			
(a)	RMSE = 3.2 pixels	(d)	RMSE = 2.2 pixels
			
(b)	RMSE = 3.5 pixels	(e)	RMSE = 3.5 pixels
			
(c)	RMSE = 3.3 pixels	(f)	RMSE = 3.6 pixels

Figure 3.10: Examples of ground-truth skeletons, and skeletons predicted using our PLS model on the VIPeR [59] data set which achieves a good skeleton fitting result.

Poor Fitting Results (VIPeR)			
Ground-Truth	PLS	Ground-Truth	PLS
(a)	RMSE = 8.8 pixels	(d)	RMSE = 7.2 pixels
(b)	RMSE = 10.7 pixels	(e)	RMSE = 9.5 pixels
(c)	RMSE = 9.1 pixels	(f)	RMSE = 11.5 pixels

Figure 3.11: Examples of ground-truth skeletons, and skeletons predicted using our PLS model on the VIPeR [59] data set which achieves a poor skeleton fitting result.

Following the experimental procedure used in other literature [43, 115], we divide the VIPeR data set randomly into two equally sized sets, with 316 identities present within each set. We conduct our experiments ten times using repeated hold-out validation, and average to produce the final result. We compare our proposed method with other state-of-the-art methods in Table 3.2, with the CMC curve shown in Figure 3.12. We can see that by using the Weighted LOMO feature descriptor (PLSAM(v1)), we achieve an improvement in all measured rank- n scores. When we concatenate the Weighted LOMO feature descriptors with the SCNCD feature descriptors at a limb-by-limb level (PLSAM(v2)), we see an even greater increase in rank- n score. We propose that this increase could be down to the vibrant clothing colours present within the VIPeR data set. The improvement in rank- n score shows how our method produces a much more robust feature descriptor for use in matching.

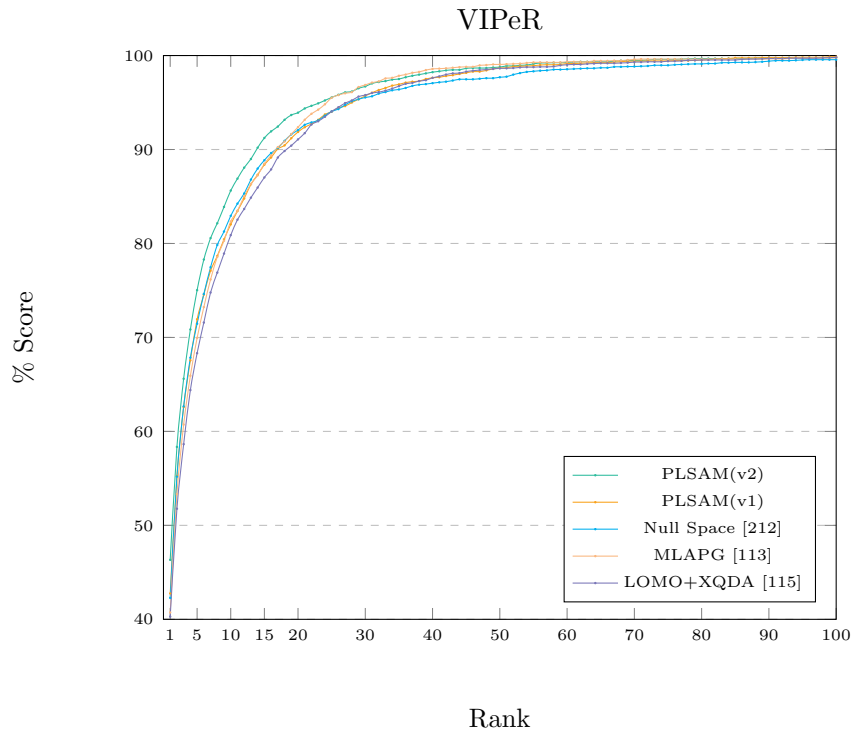


Figure 3.12: CMC on the VIPeR [59] data set. All of our CMC curves are single-shot results. Results are reproduced from [115], [217], [212], [111] and [113].

	VIPeR			
	r=1	r=5	r=10	r=20
PLSAM(v2)	46.3	75.0	85.6	93.9
PLSAM(v1)	42.8	71.9	82.0	91.9
Null Space [212]	42.3	71.5	82.9	92.1
MLAPG [113]	40.7	69.9	82.3	92.4
DeepList [184]	40.5	69.2	81.0	91.2
LOMO+XQDA [115]	40.3	68.3	80.9	91.1
SCNCD [208]	37.8	68.5	81.2	90.4

Table 3.2: The VIPeR [59] data set was split into two sets, with 316 identities allocated for training and 316 for testing. Every probe image in the test set is compared to every gallery image in the test set. PLSAM(v2) consists of the Weighted LOMO and limb-by-limb level SCNCD feature descriptors with XQDA, whereas PLSAM(v1) consists just of the Weighted LOMO feature descriptors with XQDA.

3.6.2 Evaluation on the QMUL GRID data set

In this section, we will evaluate our method using the QMUL GRID [117, 124, 125] data set. Whilst images from the VIPeR data set were all originally provided with a standard resolution of 128×48 pixels, images from the QMUL GRID data set were originally provided in a variety of different resolutions. Therefore, we begin by resizing all images to a standard resolution of 128×48 pixels. Similarly to the VIPeR data set, we extract HOG features from the V channel of HSV colour space to act as input for our skeleton prediction model. Whilst during experiments on the VIPeR data set, it was necessary to train two skeleton prediction models given the high variation of person poses present in the VIPeR data set, for the QMUL GRID data set we were able to achieve sufficiently high skeleton prediction performance using only a single skeleton prediction model. We believe this is because most people photographed as part of the QMUL GRID data set are facing either towards or away from the camera, and therefore most personal orientations are homogeneous.

Figure 3.13 shows examples of the best and worst skeleton fitting results on the QMUL GRID data set. We can see that similarly to the experimentation on the VIPeR data set, the fitting fails on those with raised arms, but performs well on those with more standard poses. We believe this can be attributed to a lack of training examples of people with raised arms. Figure 3.14 shows the distribution of RMSE on skeletons predicted using the PLS-based skeleton prediction model. Figure 3.15 shows a sample of images where our PLS-based method achieves good skeleton fitting results, whilst Figure 3.16 shows a sample of images where our PLS-based method achieves poor skeleton fitting results. We can see from Figure 3.16 that the PLS-based method struggles to accurately fit skeletons when the person has an unusual pose, such as in images (c), (e) and (f).

We use the same experimental protocols as used in various other literature [115, 126]. Therefore, as the QMUL GRID data set consists of 250 image pairs and 775 single images, we split the pairs into 125 training pairs and 125 testing pairs, and place the 775 single images into the testing gallery set. We run our experiments ten times using repeated hold-out validation, and average to produce the overall result. We present a comparison of our proposed method’s results in Table 3.3. We can see from Table 3.3 that our proposed methods greatly outperform other state-of-the-art methods. Interestingly, the improvement over the standard LOMO+XQDA method is significantly larger than that seen with experimentation on the VIPeR data set. We believe that the QMUL GRID data set benefits significantly from foreground modelling due to the high level occlusion, overlapping people, and background variation present within the data set, due to its cameras being located in a busy underground transport station. The use of foreground modelling under

these circumstances can act to mask out significant amounts of noise, computing more robust feature descriptors representative of the person. Furthermore, the large matching rate increase on both the QMUL GRID data set and the VIPeR data set demonstrates the generalisation ability of our proposed method. The CMC curve comparing our results to other state-of-the-art methods can be seen in Figure 3.17.

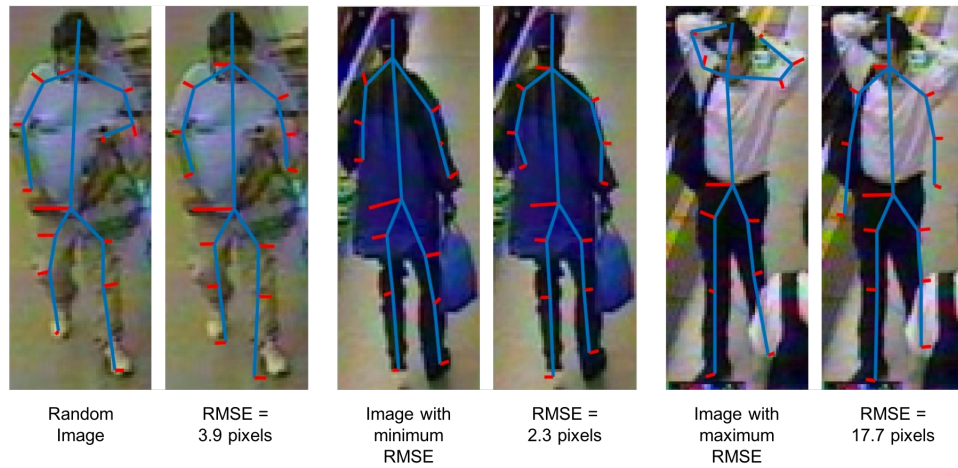


Figure 3.13: Examples of ground-truth and predicted skeleton from the QMUL GRID [117, 124, 125] data set. The average RMSE over the entire test set is 5.3 pixels.

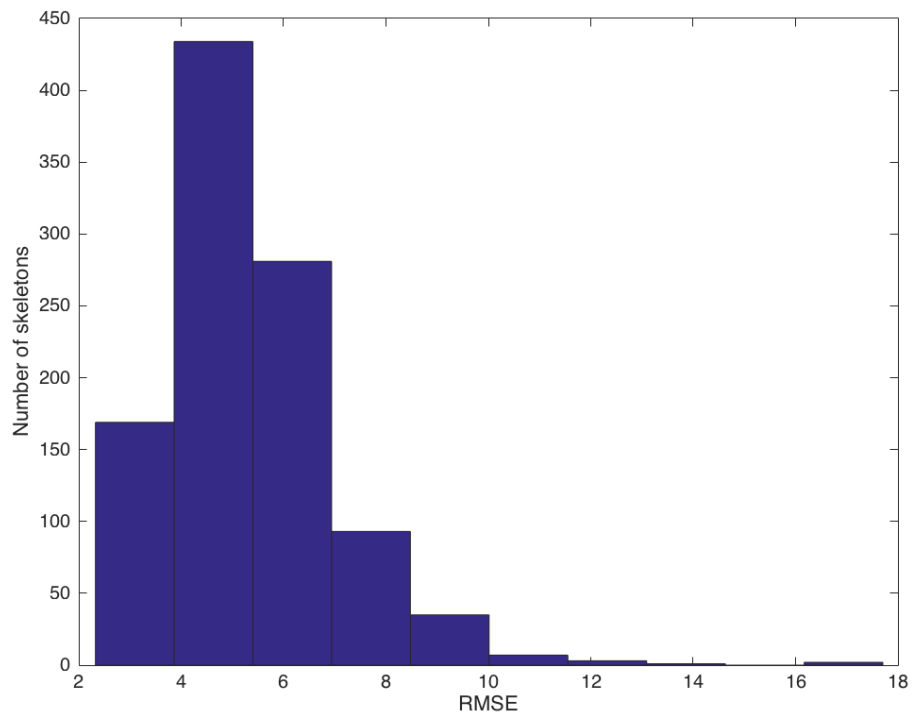


Figure 3.14: The distribution of RMSE on skeletons predicted by the PLS-based skeleton prediction model on the QMUL GRID [117, 124, 125] data set.



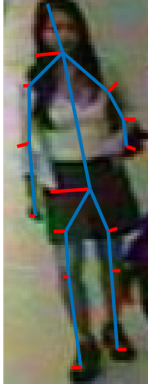
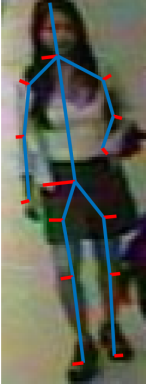





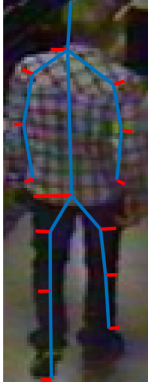
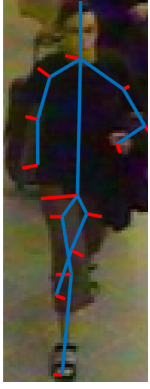

Good Fitting Results (QMUL GRID)			
Ground-Truth	PLS	Ground-Truth	PLS
			
(a)	RMSE = 3.6 pixels	(d)	RMSE = 2.5 pixels
			
(b)	RMSE = 3.5 pixels	(e)	RMSE = 2.7 pixels
			
(c)	RMSE = 3.9 pixels	(f)	RMSE = 5.1 pixels

Figure 3.15: Examples of ground-truth skeletons, and skeletons predicted using our PLS model on the QMUL GRID [117, 124, 125] data set which achieves a good skeleton fitting result.

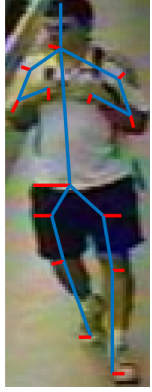






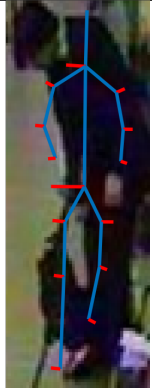




Poor Fitting Results (QMUL GRID)			
Ground-Truth	PLS	Ground-Truth	PLS
			
(a)	RMSE = 8.5 pixels	(d)	RMSE = 7.7 pixels
			
(b)	RMSE = 6.0 pixels	(e)	RMSE = 8.1 pixels
			
(c)	RMSE = 6.4 pixels	(f)	RMSE = 7.1 pixels

Figure 3.16: Examples of ground-truth skeletons, and skeletons predicted using our PLS model on the QMUL GRID [117, 124, 125] data set which achieves a poor skeleton fitting result.

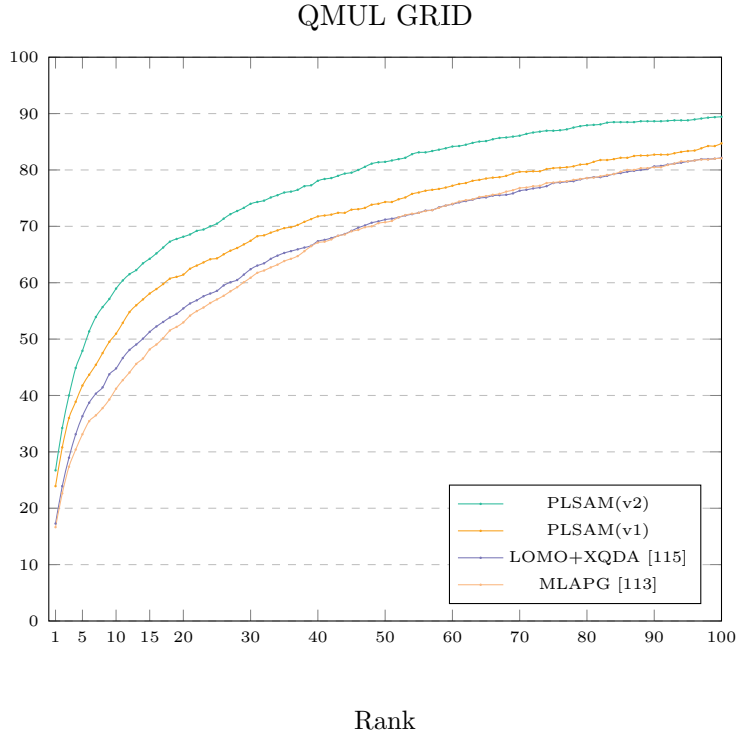


Figure 3.17: CMC on the QMUL GRID [117, 124, 125] data set. All of our CMC curves are single-shot results. Results are reproduced from [115], [217], [212], [111] and [113].

	QMUL GRID			
	r=1	r=5	r=10	r=20
PLSAM(v2)	26.7	47.9	59.0	68.2
PLSAM(v1)	23.9	41.8	51.0	61.4
MLAPG [113]	16.6	33.1	41.2	53.0
LOMO+XQDA [115]	17.3	36.3	44.8	55.4

Table 3.3: The QMUL GRID [117, 124, 125] data set was split into two sets, with 125 identities allocated for training and 900 for testing. The 900 testing identities consisted of 125 image pairs and 775 single images. Every probe image in the test set is compared to every gallery image in the test set. PLSAM(v2) consists of the Weighted LOMO and limb-by-limb level SCNCD feature descriptors with XQDA, whereas PLSAM(v1) consists just of the Weighted LOMO feature descriptors with XQDA.

3.6.3 Evaluation on the CUHK03 data set

In this section, we evaluate how our method performs on the CUHK03 [111] data set. Similarly to the QMUL GRID data set, images within the CUHK03 data set come in a variety of different resolutions, and thus we resize to 128×48 pixels for the skeleton prediction stage. However, to take advantage of the higher resolution available for most images within this data set, we also resize to 160×60 pixels for the feature extraction (LOMO and Weighted LOMO) and orientation modelling stages. We divide the data set into two distinct sets, with each distinct set containing people with roughly similar orientations. Whilst quantitative analysis could not be completed due to a lack of ground-truth skeleton data, we believe that skeletons predicted using HOG features extracted from the Y channel of the YIQ colour model appeared to be more accurate than those predicted from the V channel of the HSV colour model, and hence we chose to extract HOG features from the former. We use our aforementioned two CUHK03 orientation sets to learn a CUHK03-specific orientation model, training on 1160 identities and testing on 100 identities in line with standard experimental protocol [111, 115], and achieve an orientation accuracy of 95.8%. Again due to a lack of ground-truth skeleton keypoints for CUHK03, we instead use the skeleton prediction models as trained on VIPeR for the task of skeleton prediction, as the two data sets are quite visually similar with regards to colour vibrancy and camera viewpoints. Following orientation prediction, we predict the skeleton of each image using either the frontal or sideways VIPeR skeleton prediction models. We again follow standard experimentation protocol such as in [111, 115] by splitting the data set into 1160 training identities and 100 testing identities. We run our experiment twenty times using repeated hold-out validation, and average to produce the final result. Table 3.4 presents our results compared to competing state-of-the-art methods, where we can see that PLSAM(v1) increases the rank-1 score by 5.7%, whereas PLSAM(v2) increases the rank-1 score by 6.3%. The CMC curve can be seen in Figure 3.18.

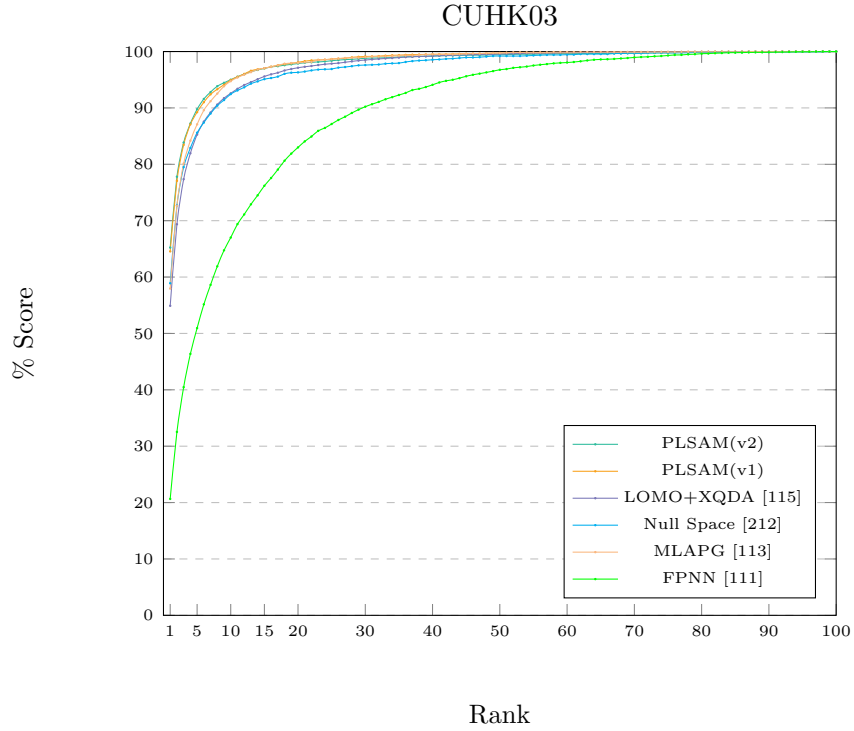


Figure 3.18: CMC on the CUHK03 [111] data set. All of our CMC curves are single-shot results. Results are reproduced from [115], [217], [212], [111] and [113].

	CUHK03			
	r=1	r=5	r=10	r=20
PLSAM(v2)	65.2	89.8	95.0	97.9
PLSAM(v1)	64.6	89.2	94.9	98.1
Null Space [212]	58.9	85.6	92.5	96.3
MLAPG [113]	58.0	87.1	94.7	98.0
DeepList [184]	55.9	86.3	93.7	98.0
LOMO+XQDA [115]	54.9	85.3	92.6	97.1
FPNN [111]	20.7	50.9	67.0	83.0

Table 3.4: The CUHK03 [111] data set was split into two sets, with 1160 identities allocated for training and 100 for testing. Every probe image in the test set is compared to every gallery image in the test set. PLSAM(v2) consists of the Weighted LOMO and limb-by-limb level SCNCD feature descriptors with XQDA, whereas PLSAM(v1) consists just of the Weighted LOMO feature descriptors with XQDA.

3.7 Summary

In this chapter, we have proposed a PLS-based method to predict the skeleton of an individual present within a Re-ID image. We train the PLS regression model by using HOG features extracted from a training set of Re-ID images, and learning a regression between the HOG features and a set of labelled skeleton keypoints. Upon predicting the skeleton of an unseen Re-ID image, we can use this information to produce a foreground mask, which provides distinction between the foreground and background regions of the Re-ID images. We then use this mask during the feature extraction stage to weight feature descriptors such that features are primarily extracted from foreground regions. Following feature extraction, we learn a distance metric which is optimal for the task of Re-ID, and prove that our method outperforms other state-of-the-art techniques.

We find that the use of a PLS regression model for foreground modelling can significantly increase the matching rate. The advantages of such a method include:

1. **Computational Efficiency:** Some methods, such as those which utilise a Deep Convolutional Neural Network (Deep CNN), require significant computational resources to learn a mapping between a set of input and output variables. With Deep CNNs, this is due to how network architectures can be large in size, and require training thousands of images multiple times over to train. Comparatively, learning a shallow regression model takes fewer computational resources.
2. **Training Requirements:** When compared to other methods, such as Deep CNNs, a shallow regression model can be trained with fewer training images. This is important given the scarcity of skeleton keypoints for Re-ID (or similar domain) imagery.
3. **Fitting accuracy of appearance modelling:** We have demonstrated that we are able to achieve a high fitting accuracy when using a PLS regression model. We believe that this is due to the simultaneous dimensionality reduction and distance metric learning states employed by the PLS regression technique.

Other methods, such as Canonical Correlation Analysis (CCA) or a multilayer perceptron, may also be able to predict accurate skeletons, and could form the basis for future work.

Furthermore, our proposed Weighted LOMO descriptor also possesses the following advantages:

1. **Easy integration with the foreground mask:** Once the PLS regression model has predicted the skeleton keypoints of an unseen input image, these skeleton key-

points can be used to produce a foreground mask, providing visual distinction between the foreground and background regions. As the original LOMO feature descriptors are extracted in patches, we can easily weight feature descriptors on a patch-by-patch level by incorporating the information present within the foreground mask.

2. Invariance to small errors in skeleton prediction: It is difficult to compute a perfectly accurate skeleton fit on all occasions. This is especially prevalent in Re-ID imagery, where pose and illumination variation, low resolution, and occlusion can make predicting the skeleton of an individual even more difficult. In our proposed method, instead of only extracting features from foreground regions, we weight each patch according to the percentage of each patch considered foreground. Due to this, foreground regions erroneously classified as background may not be excluded from the feature extraction process.

Recently [2, 83, 108, 111, 112, 193], methods utilising Deep CNNs have shown great promise in both regression tasks and for use in the field of Re-ID. However, aforementioned issues specifically relating to computational requirements and the magnitude of training data required to train such a network still remain. In the following chapter, we will show how we overcome these issues to create a Deep CNN for use as a skeleton prediction model.

Chapter 4

Deep Foreground Appearance Modelling

4.1 Introduction

In the previous chapter, we introduced a PLS-based approach to learning a regression between image appearance information and skeleton keypoints. Skeletons predicted through the PLS-based model could then be used within the proposed Re-ID method to permit feature weighting in areas considered by the model to be foreground. We demonstrated an increase in Re-ID matching rates following the inclusion of foreground modelling within the pipeline. However, it was necessary to train multiple PLS models due to the poor generalisation ability of these models when presented with images of people taken from different orientations, such as frontal images compared to sideways images.

In this chapter, we propose to instead replace the PLS component of our framework with a deep convolutional neural network (CNN)-based approach [190]. This approach enables our model to learn intricate poses through network training using a large variety of different poses. By using a deep CNN rather than a PLS-based model, we are able to create a skeleton prediction model able to predict person skeletons to a high degree of accuracy regardless of the orientation of the person. In Section 4.2 we describe our deep neural-network appearance model (DNAM), including detail of the network architecture. Section 4.3 compares our results against other competing methods, whilst we summarise our findings in Section 4.4.

4.2 Deep Neural-Network Appearance Modelling

In this section, we will discuss how we design a deep CNN to learn a mapping between input images and skeleton keypoints.

We define a CNN architecture which takes as input a Re-ID image, and outputs the corresponding skeleton keypoint locations. Our CNN is based on the ResNet-50 [66] architecture, which has demonstrated high performance in computer vision tasks in recent years [68, 116, 228, 231]. The use of a ResNet-50 architecture allows us to take advantage of transfer learning [147] by using weights pre-trained on the ImageNet [38] data set. By utilising transfer learning, we are able to decrease the time taken for the network to converge, and thus speed up the training process. As the use of the pre-trained ImageNet weights mandates a specific CNN architecture, we resize all input images to 224×224 pixels to meet the required size of the input layer. Given the difference in application between the network on which the ImageNet weights were trained, and our network, we remove the final fully-connected layers and replace with our own fully-connected layers of size 1024 and 58 respectively, where 58 is the dimensionality of our skeleton vector (29 x -coordinates and 29 y -coordinates). Hence, each unit in the final fully-connected layer of our network represents an x or y co-ordinate in the skeleton vector. We use the hand-labelled skeleton keypoints initially described in Appendix A. To summarise, the skeletons consist of fourteen limbs each represented by two end-points, as well as a third keypoint representing the edge of the limb. The distance between the bottom end-point and the keypoint representing the edge of the limb allow us to calculate the width of the limb. In addition, the bottom keypoint of each limb is also the top keypoint of the following limb. Figure 4.1 demonstrates our network’s architecture.

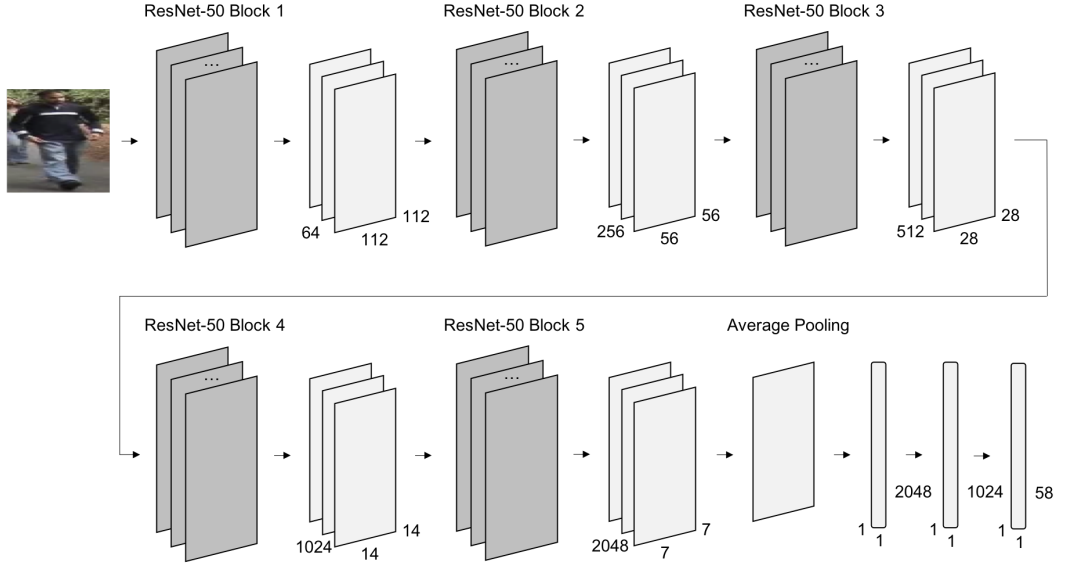


Figure 4.1: The network architecture for our proposed deep foreground modelling network. We first rescale images to a resolution of 224×224 pixels, and pass the images through the convolutional layers of a ResNet-50 network [66]. We take the POOL5 average pooling layer as the output of the ResNet-50 model, flatten the output, and pass the output through two further fully-connected layers. The output of our proposed network contains 58 units, representing the (x, y) coordinates of the skeleton key-points (joints and edge markers). We use the RMSProp [177] optimizer, with a mean squared error loss.

For each given Re-ID data set or set of data sets, we first designate a subset of these identities as a training set. However, typical Re-ID data sets are small in size and lack significant numbers of examples per class (identity), and therefore, given deep networks require a significant number of training examples, training a deep CNN can be difficult. To expand the size of our training and validation sets, we apply data augmentation to all images. To achieve this, we create additional images and corresponding skeletons by applying small rotations and translations, as well as reflections in the y-axis. Examples of images, skeletons and their corresponding augmentations can be seen in Figure 4.2.



Figure 4.2: Examples of images, skeletons, and their corresponding augmentations. The first image in each row is the original image. The remaining images in each row are augmentations of the first image.

Unlike the PLS-based model described in Section 3, we only train a single skeleton prediction model, rather than training multiple models to handle multiple person orientations. However, similarly to Section 3.4, we extract feature descriptors from the Re-ID images using the Weighted LOMO and limb-by-limb level SCNCD feature types, which creates a feature descriptor consisting of colour and textural information weighted towards the foreground regions of a Re-ID image.

4.3 Results and Discussion

We experiment on the following three data sets: VIPeR [59], QMUL GRID [117, 124, 125], and CUHK03 [111]. For all three data sets, we begin by training the fully-connected layers of our ResNet-50-based [66] CNN architecture. This is because the non-fully-connected layers are initialised with the ImageNet [38] weights. Fine-tuning both the randomly-assigned weights of the fully-connected layers and the pre-trained on ImageNet weights of the remaining model would negatively impact the pre-trained weights. Following training on the fully-connected layers, we train all layers from ResNet-50’s third stage onwards. We use the RMSProp [177] optimizer with a learning rate of 0.001.

4.3.1 Evaluation on the VIPeR data set

We divide the VIPeR [59] data set into two distinct sets. The first set consists of 316 identities for training/validation, whilst the second also consists of 316 identities and is used for testing. 80% of the training/validation identities are designated as training identities, with the remaining as validation identities. In addition to the VIPeR data set, we further supplement the training and validation sets with all images from the QMUL GRID [117, 124, 125] data set, due to the large amount of data required when using deep learning approaches when compared to more traditional approaches. The QMUL GRID [117, 124, 125] data set is firstly split into two distinct sets; The first set contains the images which form an image pair, that being, there is greater than a single image of these identities present within the unaugmented data set. The second set contains the images which do not form an image pair, and hence only one image of this identity is present within the unaugmented data set. We proceed by taking 80% of identities from each set to complement the training set, with the remaining 20% forming part of the validation set. Splitting the QMUL GRID data set this way ensures that we use a consistent number of training and validation images between folds. We train for 15 epochs, with a batch size of 32. Although the use of the ImageNet [38] mandates images of size 224×224 during the skeleton prediction stage, we resize all images to 128×48 for feature extraction, also

scaling the predicted skeletons accordingly. Examples of skeleton prediction on the VIPeR data set using our deep method can be seen in Figure 4.3. Figure 4.4 shows the distribution of RMSE on skeletons predicted using the deep skeleton prediction model on the VIPeR data set. Figure 4.5 shows a sample of images where the method achieves good skeleton fitting results, and compares the fitting to the PLS-based approach from Chapter 3. In Figure 4.5, we can see that the approach better fits the person in (a)’s left hand, even though it is a similar colour to the background. A similar situation can be seen in person (b), where a better fitting is obtained on the legs, which are a similar colour to the shadowed region in the background. In (c), the person is in a common pose, with their back to the camera, and both approaches achieve a similar RMSE. Figure 4.6 shows a sample of images where the method achieves poor skeleton fitting results, and also compares the fitting to the PLS-based approach from Chapter 3. We can see in person (a) that both the deep CNN-based approach and the PLS-based approach struggle to fit the skeleton, and we believe this is due to the unusual coloured clothing that person (a) is wearing on their upper body. In person (b), both approaches fail to fit person’s left arm appropriately, which is raised. Person (c) contains significant illumination and blur issues, and both approaches fail to identify this person as standing perpendicular to the camera, instead fitting a frontways skeleton. In Figure 4.7 and Figure 4.8, we compare the fittings achieved by the method with our PLS-based method on the images which had fittings considered poor within Figure 3.11. We can see that in all but one case, the CNN-based approach achieves a lower RMSE. In (b), we can see that the fitting of the legs is considerably better, whilst in (c), (e) and (f), it better captures the orientation of the person relative to the camera.





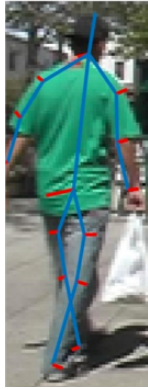



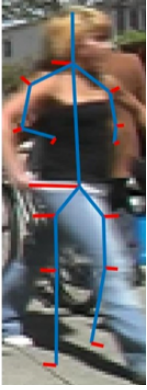
Ground-Truth	DNAM	PLS
		
Random Image	RMSE = 6.5 pixels	RMSE = 7.6 pixels
		
Image with minimum RMSE	RMSE = 1.5 pixels	RMSE = 2.9 pixels
		
Image with maximum RMSE	RMSE = 17.9 pixels	RMSE = 12.3 pixels

Figure 4.3: Examples of ground-truth, skeletons predicted using our deep model, and skeletons predicted using our PLS model on the VIPeR [59] data set. The average RMSE when using the deep model was 4.5 pixels, whilst the average when using the PLS model was 5.2 pixels.

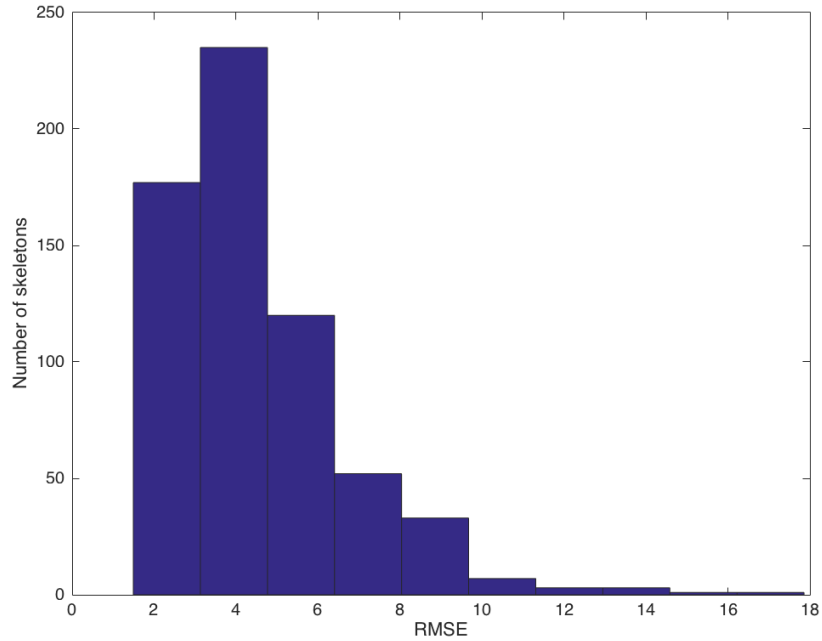


Figure 4.4: The distribution of RMSE on skeletons predicted by the deep skeleton prediction model on the VIPeR [59] data set. The average RMSE was 4.5 pixels, whilst the average using the PLS model was 5.2 pixels.

For the XQDA distance metric learning stage, we combine the training and validation sets to form a larger training set. This is because the XQDA distance metric learning stage does not require a validation set at this stage, as we use the same hyperparameters as used in Chapter 3. We repeat our experiment ten times using repeated hold-out validation, and average to produce the final result. We can see from Table 4.1 that our deep neural-network appearance modeling (DNAM) model performs better than all other methods, including the PLSAM method [189] as proposed in Chapter 3, at rank-10 and rank-20. However, the PLSAM method outperforms our DNAM method at rank-1 and rank-5. When compared to using only LOMO features, we can see an increase of 5.0% in the rank-1 rate when using our DNAM(v2) method, but we also see a 1.0% decrease when compared to our










Good Fitting Results (VIPeR)		
Ground-Truth	DNAM	PLS
		
(a)	RMSE = 3.0 pixels	RMSE = 4.3 pixels
		
(b)	RMSE = 3.8 pixels	RMSE = 5.2 pixels
		
(c)	RMSE = 3.6 pixels	RMSE = 3.3 pixels

Figure 4.5: Examples of ground-truth, skeletons predicted using our deep model, and skeletons predicted using our PLS model on the VIPeR [59] data set which achieve a good skeleton fitting result.










Poor Fitting Results (VIPeR)		
Ground-Truth	DNAM	PLS
		
(a)	RMSE = 7.8 pixels	RMSE = 5.7 pixels
		
(b)	RMSE = 9.1 pixels	RMSE = 8.4 pixels
		
(c)	RMSE = 10.1 pixels	RMSE = 12.9 pixels

Figure 4.6: Examples of ground-truth, skeletons predicted using our deep model, and skeletons predicted using our PLS model on the VIPeR [59] data set which achieve a poor skeleton fitting result.

DNAM versus PLS poor fitting results (VIPeR)		
Ground-Truth	DNAM	PLS
(a)	RMSE = 9.0 pixels	RMSE = 8.8 pixels
(b)	RMSE = 8.8 pixels	RMSE = 10.7 pixels
(c)	RMSE = 5.8 pixels	RMSE = 9.1 pixels

Figure 4.7: A comparison of skeleton fitting results using our deep method against the fittings labelled as poor in Figure 3.11.










DNAM versus PLS poor fitting results (VIPeR)		
Ground-Truth	DNAM	PLS
		
(d)	RMSE = 7.1 pixels	RMSE = 7.2 pixels
		
(e)	RMSE = 4.9 pixels	RMSE = 9.5 pixels
		
(f)	RMSE = 9.3 pixels	RMSE = 11.5 pixels

Figure 4.8: Further comparisons of skeleton fitting results using our deep method against the fittings labelled as poor in Figure 3.11.

PLSAM(v2) method. We believe the cause of this may be the similarity between the skeletons predicted by the PLS and deep models, even when considering that the deep method had a lower average root mean squared error. The CMC curve can be seen in Figure 4.9.

	VIPeR			
	r=1	r=5	r=10	r=20
DNAM(v2)	45.3	74.8	86.4	94.2
DNAM(v1)	42.3	71.4	81.9	91.8
PLSAM(v2) [189]	46.3	75.0	85.6	93.9
PLSAM(v1) [189]	42.8	71.9	82.0	91.9
DeepDiff [77]	43.2	68.0	77.6	86.1
Null Space [212]	42.3	71.5	82.9	92.1
MLAPG [113]	40.7	69.9	82.3	92.4
DeepList [184]	40.5	69.2	81.0	91.2
LOMO+XQDA [115]	40.3	68.3	80.9	91.1
SCNCD [208]	37.8	68.5	81.2	90.4
PKFM [23]	36.8	70.4	83.7	91.7
CSBT [26]	36.6	66.2	-	88.3
DCML [132]	33.6	62.9	76.5	87.6

Table 4.1: Results on the VIPeR [59] data set. The best results are highlighted in bold. (v1) refers to the Weighted LOMO features and XQDA, whereas (v2) refers to the Weighted LOMO features with the limb-by-limb level SCNCD features and XQDA.

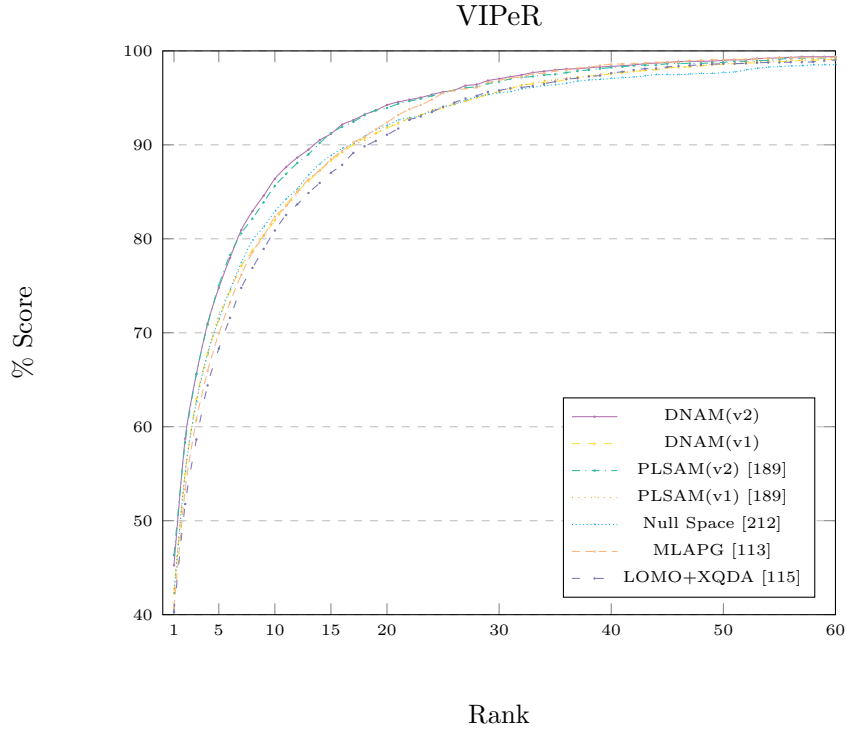


Figure 4.9: CMC on the VIPeR [59] data set. All of our CMC curves are single-shot results. Results are reproduced from [77, 111, 113, 115, 189, 212, 217].

4.3.2 Evaluation on the QMUL GRID data set

We split the QMUL GRID [117, 124, 125] data set into two sets. The first set contains 125 identities taken from the identities which form an image pair, and are used for training/validation. The second set contains the remaining 125 identities which form an image pair, and are used for testing. Regarding the 775 identities/images which do not form an image pair, we use these to increase the size of the testing gallery set. We enlarge the training/validation set by also including the entirety of the VIPeR [59] data set, with 80% of the VIPeR data set being used for training, with the remaining 20% being used for validation. Similarly to our experimentation on the VIPeR data set (Section 4.3.1), we combine the training and validation sets for the XQDA distance metric learning stage. We train for 10 epochs, and set the batch size for 16. We resize the images to 128×48 pixels for the feature extraction stage, and rescale the skeletons appropriately. We run our experiments ten times using repeated hold-out validation, and average to produce the final result. Examples of skeleton prediction on the QMUL GRID data set using our deep method can be seen in Figure 4.10. Figure 4.11 shows the distribution of RMSE on skeletons predicted using the deep skeleton prediction model on the QMUL GRID data set. Figure 4.12

shows a sample of images where our deep CNN-based method achieves good skeleton fitting results, and compares the fitting to the PLS-based approach from Chapter 3. The QMUL GRID data set is easier to perform skeleton fitting when compared to the VIPeR data set, as most images are taken from the front or behind of each person, hence reducing the need for the skeleton fitting model to handle different person orientations. In (a), we can see that the PLS-based model mistakenly considers a bag to be a person’s legs, whereas this mistake is not made by the deep CNN-based model. In (c), the deep CNN-based model fits the raised left arm of the person to a higher accuracy than the PLS-based model. Figure 4.13 shows a sample of images where our deep CNN-based method achieves poor skeleton fitting results, and also compares the fitting to the PLS-based approach from Chapter 3. In person (a), both the deep CNN-based approach and the PLS-based approach struggle to fit the raised left arm of the person, most likely due to a lack of training data for this pose. (b) and (c) both contain occlusion, and hence both the PLS-based and deep CNN-based approaches struggle to appropriately fit the legs. In Figure 4.14 and Figure 4.15, we compare the fittings achieved by our deep CNN-based method with our PLS-based method on the images which had fittings considered poor within Figure 3.16. We can see that in all but one case, the deep CNN-based approach achieves a lower RMSE. In (a), the deep CNN-based approach better fits the person’s raised arms. In (b), the deep CNN-based approach better fits the person’s legs, which are a very similar colour to part of the background. Both images (c) and (d) contain occlusion, and hence both approaches struggle to fit the legs. In (e), the deep CNN-based approach better fits the orientation of the individual, even though this orientation is rare within the training set.

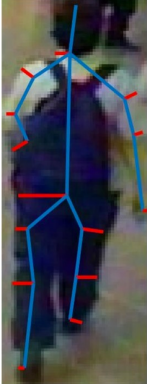
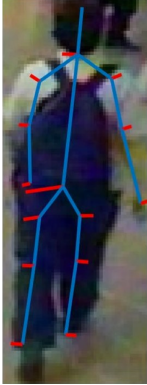







Ground-Truth	DNAM	PLS
		
Random Image	RMSE = 4.3 pixels	RMSE = 4.5 pixels
		
Image with minimum RMSE	RMSE = 2.2 pixels	RMSE = 3.9 pixels
		
Image with maximum RMSE	RMSE = 18.3 pixels	RMSE = 17.6 pixels

Figure 4.10: Examples of ground-truth, skeletons predicted using our deep model, and skeletons predicted using our PLS model on the QMUL GRID [117, 124, 125] data set. The average RMSE when using the deep model was 5.5 pixels, whilst the average when using the PLS model was 5.3 pixels.

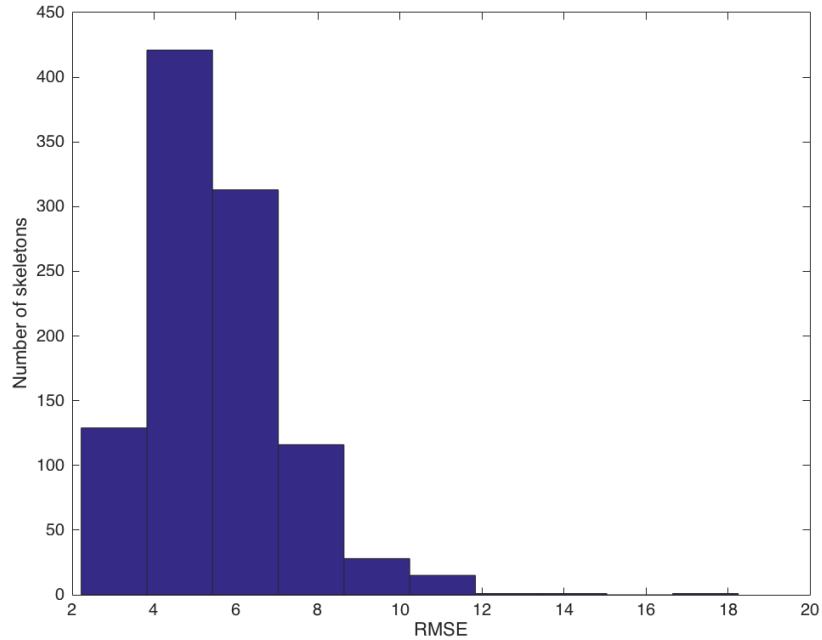


Figure 4.11: The distribution of RMSE on skeletons predicted by the deep skeleton prediction model on the QMUL GRID [117, 124, 125] data set. The average RMSE was 5.5 pixels, whilst the average using the PLS model was 5.3 pixels.

From Table 4.2, we can see that our proposed deep neural-network appearance model gives the highest rank- n results across all presented values of n . Compared to using simply the original LOMO features, we achieve an increase of 11.1% in the rank-1 rate. When comparing the PLSAM(v2) result to our proposed DNAM(v2), we see an increase in 1.7% increase in the rank-1 rate. Figure 4.10 shows that the average root mean squared error for both the deep neural-network method and the PLS-based method are similar, with only 0.2 pixels between them. The CMC curve for the experiments on the QMUL GRID data set can be seen in Figure 4.16.

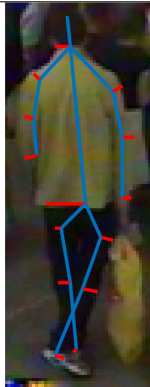
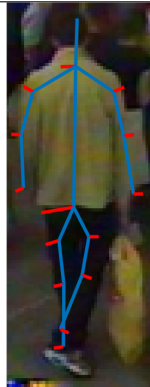







Good Fitting Results (QMUL GRID)		
Ground-Truth	DNAM	PLS
		
(a)	RMSE = 4.1 pixels	RMSE = 5.0 pixels
		
(b)	RMSE = 3.0 pixels	RMSE = 3.4 pixels
		
(c)	RMSE = 2.8 pixels	RMSE = 4.8 pixels

Figure 4.12: Examples of ground-truth, skeletons predicted using our deep model, and skeletons predicted using our PLS model on the QMUL GRID [117, 124, 125] data set which achieve a good skeleton fitting result using our deep model.










Poor Fitting Results (QMUL GRID)		
Ground-Truth	DNAM	PLS
		
(a)	RMSE = 8.1 pixels	RMSE = 9.6 pixels
		
(b)	RMSE = 11.4 pixels	RMSE = 10.1 pixels
		
(c)	RMSE = 8.2 pixels	RMSE = 6.8 pixels

Figure 4.13: Examples of ground-truth, skeletons predicted using our deep model, and skeletons predicted using our PLS model on the QMUL GRID [117, 124, 125] data set which achieve a poor skeleton fitting result using our deep model.









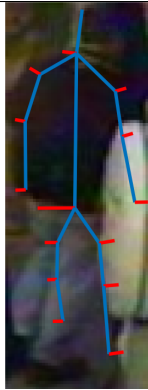
DNAM versus PLS poor fitting results (QMUL GRID)		
Ground-Truth	DNAM	PLS
		
(a)	RMSE = 6.4 pixels	RMSE = 8.5 pixels
		
(b)	RMSE = 5.8 pixels	RMSE = 6.0 pixels
		
(c)	RMSE = 5.0 pixels	RMSE = 6.4 pixels

Figure 4.14: A comparison of skeleton fitting results using our deep method against the fittings labelled as poor in Figure 3.16.


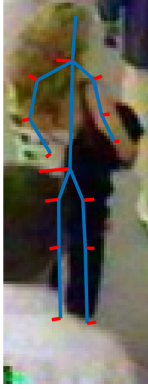
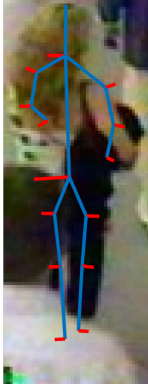






DNAM versus PLS poor fitting results (QMUL GRID)		
Ground-Truth	DNAM	PLS
		
(d)	RMSE = 5.7 pixels	RMSE = 7.7 pixels
		
(e)	RMSE = 6.9 pixels	RMSE = 8.1 pixels
		
(f)	RMSE = 8.2 pixels	RMSE = 7.1 pixels

Figure 4.15: Further comparisons of skeleton fitting results using our deep method against the fittings labelled as poor in Figure 3.16.

	QMUL GRID			
	r=1	r=5	r=10	r=20
DNAM(v2)	28.4	49.2	60.0	68.8
DNAM(v1)	24.3	41.6	52.4	61.8
PLSAM(v2) [189]	26.7	47.9	59.0	68.2
PLSAM(v1) [189]	23.9	41.8	51.0	61.4
MLAPG [113]	16.6	33.1	41.2	53.0
LOMO+XQDA [115]	17.3	36.3	44.8	55.4
PKFM [23]	16.3	35.8	46.0	57.6

Table 4.2: Results on the QMUL GRID [117, 124, 125] data set. The best results are highlighted in bold. (v1) refers to the Weighted LOMO features and XQDA, whereas (v2) refers to the Weighted LOMO features with the limb-by-limb level SCNCD features and XQDA.

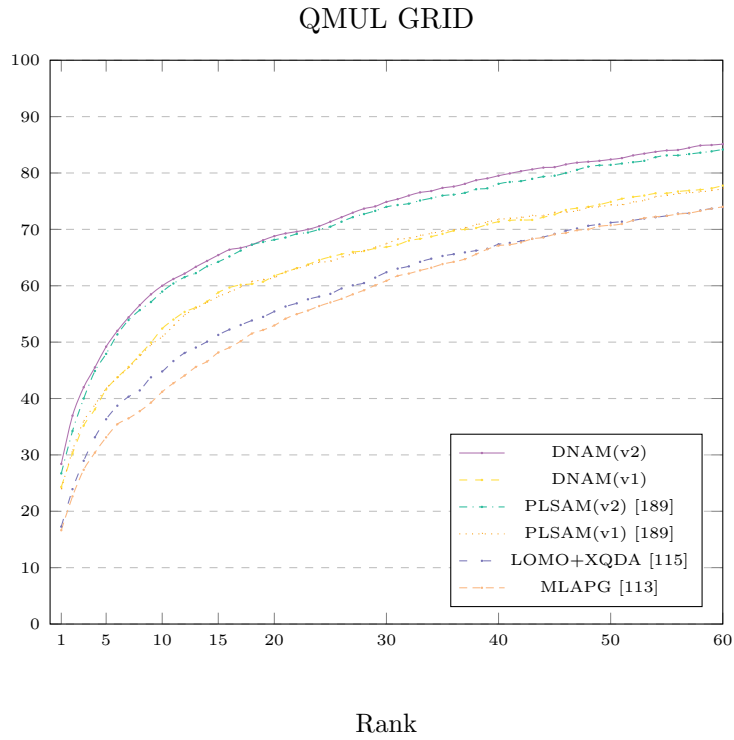


Figure 4.16: CMC on the QMUL GRID [117, 124, 125] data set. All of our CMC curves are single-shot results. Results are reproduced from [77, 111, 113, 115, 189, 212, 217].

4.3.3 Evaluation on the CUHK03 data set

We use the manually cropped version of the CUHK03 [111] data set, and split the data into 1160 training identities and 100 testing identities. As we do not have any ground-truth, hand-labelled skeletons for the CUHK03 data set, we instead train our deep model using the entirety of the VIPeR and QMUL GRID data sets. We divide the VIPeR and QMUL GRID data sets into training, validation and testing sets as described in the previous two sections. We train our model for 15 epochs, with a batch size of 32. Similarly to the feature extraction in the previous (Section 3.6.3), we scale the images within the CUHK03 data set to 160×60 for extracting the Weighted LOMO feature descriptors. We run our experiments twenty times using repeated hold-out validation, and average to produce the final results. From Table 4.3, we can see that the PLS-based method outperforms the deep method. We believe that this can be explained by the PLS-based method generalising better when an unseen image from an unseen data set is passed to the model. Furthermore, person orientations present within the VIPeR and QMUL GRID data set are fairly constant, with people largely photographed from the front or back. This is in contrast to the CUHK03 data set, where roughly half of the images consist of those photographed from the front or back, whilst the other half consist of those photographed from the side. The use of a single model in the deep neural-network based method versus the use of two models used in the PLS-based method leads to a frontal skeleton being fit more often than in the PLS-based method. The CMC curve representing experiments on the CUHK03 data set can be seen in Figure 4.17.

	CUHK03			
	r=1	r=5	r=10	r=20
DNAM(v2)	62.2	88.0	94.2	97.5
DNAM(v1)	61.9	88.2	94.2	97.5
PLSAM(v2) [189]	65.2	89.8	95.0	97.9
PLSAM(v1) [189]	64.6	89.2	94.9	98.1
DeepDiff [77]	62.4	87.9	93.6	96.7
Null Space [212]	58.9	85.6	92.5	96.3
MLAPG [113]	58.0	87.1	94.7	98.0
DeepList [184]	55.9	86.3	93.7	98.0
CSBT [26]	55.5	84.3	-	98.0
LOMO+XQDA [115]	54.9	85.3	92.6	97.1
FPNN [111]	20.7	50.9	67.0	83.0

Table 4.3: Results on the CUHK03 [111] data set. The best results are shown in bold. (v1) refers to the Weighted LOMO features and XQDA, whereas (v2) refers to the Weighted LOMO features with the limb-by-limb level SCNCD features and XQDA.

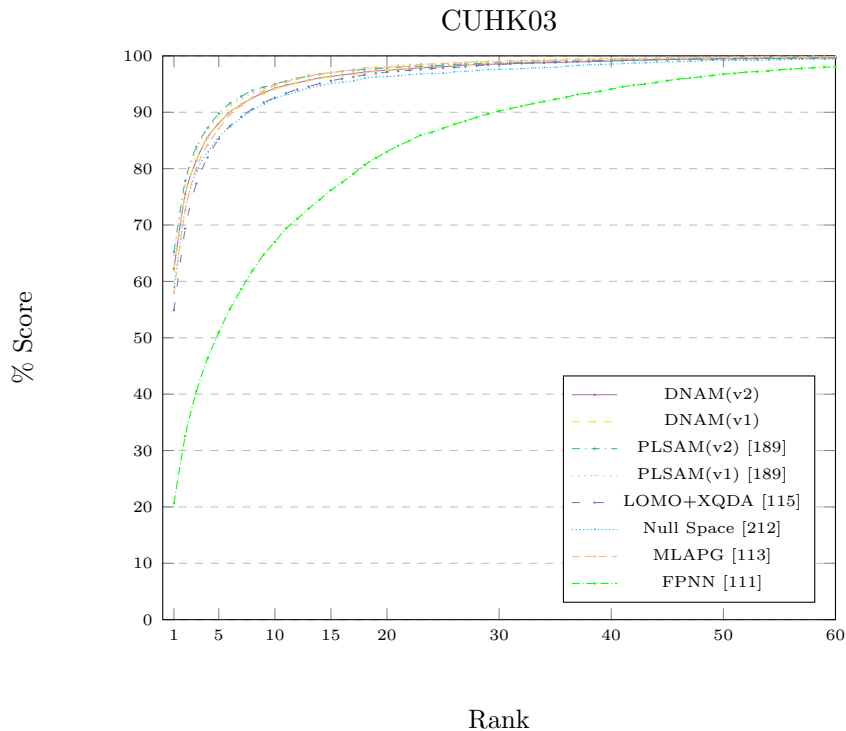


Figure 4.17: CMC on the CUHK03 [111] data set. All of our CMC curves are single-shot results. Results are reproduced from [77, 111, 113, 115, 189, 212, 217].

4.4 Summary

In this chapter, we have proposed a deep CNN-based approach called Deep Neural-Network Appearance Model (DNAM), which was used to predict the skeleton of a person in a Re-ID image. We first divide our data set into training, validation and testing sets, and then apply data augmentation to increase the size of the training and validation sets. These images, as well as their corresponding hand-labelled skeleton keypoints, are then fed into the network which learns a regression between the two. We then follow with the Weighted LOMO and limb-by-limb level SCNCD feature extraction processes, XQDA distance metric learning, and evaluation.

We compare our results with those achieved through the PLS-based approach proposed in Chapter 3, evaluating both through the Re-ID matching framework. We have demonstrated that by using a deep approach, accurate skeletons can be predicted with a single deep CNN model. We have demonstrated that when using our deep foreground modelling approach in combination with feature weighting, we can achieve superior matching performance. Experiments on the VIPeR, QMUL GRID and CUHK03 data sets have shown that the deep neural-network approach achieves an increase in the rank-1 rate of 5%, 11.1% and 7.3% in each respective

data set when compared to standard approached using the original LOMO features. When including both the PLS-based approach as proposed in Chapter 3 and our deep neural-network approach, we are able to achieve an increase of 6%, 11.1% and 10.3%.

The advantages of our deep neural-network method are:

1. Fitting accuracy: We have demonstrated that we are able to achieve a high fitting accuracy when using a deep CNN-based skeleton prediction model. When compared to the PLS-based model as proposed in Chapter 3, we have shown that we are able to achieve a greater skeleton fitting accuracy on the VIPeR data set, and comparable skeleton fitting results on the QMUL GRID data set.
2. Number of skeleton prediction models required: Whilst the PLS-based method required separate models to be trained to handle significantly different person orientations, we were able to train a single deep CNN-based model which was able to handle people of a variety of different orientations due to its nonlinearity.
3. Generalisation between data sets for skeleton prediction: When evaluating on the VIPeR data set, in addition to training on the VIPeR data sets training set, we enlarged the training set by also including the QMUL GRID data set. Similarly, when evaluating on the QMUL GRID data set, we enlarged the training set by incorporating the VIPeR data set. Through this, we were able to increase the size of the training sets, leading to the model converging to an accurate skeleton prediction result faster. In addition, due to having no ground-truth, hand-labelled skeleton keypoints for the CUHK03 data set, we trained our skeleton prediction model using only images and corresponding skeleton keypoints from the VIPeR and QMUL GRID data sets. Even so, we were able to achieve good skeleton prediction results on the CUHK03 data set. Given CNNs require a significant amount of training data to be accurate, it is not uncommon to see multiple smaller data sets with significant variation in visual characteristics, merged to form a much larger training set. Hence, such generalisation is of great importance.
4. Incorporation of transfer learning: Whilst the PLS-based models proposed in Chapter 3 were trained from scratch, the CNN-based approach was able to incorporate a ResNet-50 [66] CNN architecture using pre-trained weights trained on the ImageNet [38] data set. Therefore, we have shown that skeleton prediction for Re-ID images can utilise transfer learning to produce accurate skeleton fitting results.

However, it is important to note that whilst impressive skeleton fitting results have been observed when using the deep approach compared to when using the PLS-based approach, the deep approach did benefit from an expanded training set when compared to the PLS-based approach. Further experimentation could better standardise the experimental settings between the two approaches to provide a more thorough comparison.

Whilst this chapter has utilised deep CNNs for skeleton prediction, the feature descriptors used for the task of Re-ID are still entirely hand-crafted. However, recent work [2, 27, 28, 51, 90, 108, 111, 112, 200, 203, 206, 229, 230] has instead incorporated deep features, which are optimal to the task in which they are used. Some methods [2, 51, 111, 200, 229, 230] build deep CNNs which learn features optimal for the task of predicting whether a pair of images are of the same identity or otherwise, whereas others [51, 108, 112, 203, 206, 229, 230] instead focus on the task of predicting the identity of a person within a given image. However, significant variation in these images, such as illumination variation, can hinder the performance of these methods. An alternative [90, 101, 164, 170, 171, 174] approach is to train a deep CNN capable of detecting person attributes, such as hair length, shirt colour, and gender, which is analogous to the approach to this task taken by a human. These feature types are intrinsically more invariant to changes in visual characteristics, such as illumination and pose. In the following chapter, we will train an attribute recognition network, where the penultimate layer within the network is used in combination with hand-crafted features to act as a feature descriptor during matching.

Chapter 5

Combining Deep Features and Attribute Detection for Re-ID

5.1 Introduction

In Chapter 4, we introduced a deep CNN-based method to predict the skeleton of a person in a Re-ID image, replacing the PLS-based approach proposed in Chapter 3. The model generated by the deep CNN method was able to better predict skeletons of people photographed from a wide variety of orientations, without the need to train multiple models. Following the skeleton prediction stage, the skeletons were used to generate a binary feature map, which allowed feature descriptors to be extracted from primarily foreground regions. These feature descriptors were based on the LOMO [115] and SCNCD [208] feature extraction methods, hand-crafted features which extract colour and textural information. As such, significant visual variations between images, such as illumination variation or difference in blur, could lead to great difference in the colour and textural information extracted from each image.

Whilst traditional, automated Re-ID methods have routinely used appearance features to describe Re-ID images, humans instead rely on attribute descriptions, such as *short sleeves*, *brown hair* and *wearing a necklace* to describe a person. Figure 2.11 shows examples of attributes, as well as positive and negative examples of Re-ID images for each of those attributes.

Attribute features have an advantage over appearance features by having greater invariance to illumination and pose variation. For example, a *blue shirt* will still be considered a *blue shirt* by a human even after significant variation in illumination and pose. Figure 5.1 shows an example of numerous Re-ID images which contain people wearing blue clothing on their upper bodies according to labels provided by the PETA [39] data set, even though there is significant visual variation

between the images.



Figure 5.1: An example of images from the VIPeR [59] data set. All of these images are labelled as wearing blue clothing on their upper bodies by the PETA [39] data set. However, significant visual variation can be seen between images, such as pose and illumination variation.

Attribute detection networks can be trained which take a series of input images, and predict the presence of a set of attributes within the images. Given the presence of attributes such as *wearing glasses* and *brown shoes*, are highly correlated with image regions, spatial modeling is often used to aid attribute prediction [100, 101, 121]. This allows an attribute detection network to determine which regions of a Re-ID image are informative for the prediction of each specific attribute.

In this chapter, we propose the combination of our deep CNN-based skeleton prediction as detailed in Chapter 4 with an attribute prediction network. Our proposed attribute detection network takes four images as input: the whole image, and three parts images divided according to the predicted skeleton. We then pass these images to a network consisting of four ResNet-50-based [66] deep CNNs, concatenating the output of each sub-network as a feature descriptor. The output feature descriptor is then passed to a fully-connected layer with n nodes, with n being the number of attributes being predicted.

This chapter consists of three contributions: The first is a pose-informed attribute detection network which learns a mapping between the four input images, and an attribute vector. The robust deep attribute feature extracted from this network can then be used to produce high matching rates. The second contribution is the combination of our deep attribute feature with the LOMO [115] feature

descriptor, which improve matching rates even further. The third and final is the use of a Weighted Binary Cross Entropy (WBCE) function which weights the cost of a positive error relative to a negative error, depending on the ratio of positive to negative instances of each attribute. Our generated feature descriptors are then used in combination with the XQDA distance metric learning technique [115] to perform matching. We evaluate our method on the VIPeR [59], PRID2011 [70], i-LIDS [57, 225, 226] and Market-1501 [219] data sets, and demonstrate competitive performance against other state-of-the-art methods.

5.2 Deep Attribute Prediction

In this section, we describe our attribute prediction model, which takes four images as input, and predicts the presence or absence of a set of attributes. We utilise this model as a feature extractor, and follow by performing matching.

5.2.1 Deep Attribute Prediction Network

Using the deep CNN-based skeleton predictor proposed in Chapter 4, we divide each Re-ID image into three parts. The top part represents the head and shoulders, the middle part consists of the torso and arms, whilst the third and final part consists of the legs. As the skeleton is estimated, we extend the bounding box by 15% in the x and y dimensions, to account for skeleton prediction errors. We take the original image, as well as the three body parts images, and resize each image to a resolution of 224×224 pixels, and apply the standard ResNet-50 [66] preprocessing algorithms. Figure 5.2 shows an example of how we divide a Re-ID image into three parts using the predicted skeleton.

Each attribute feature is represented as a binary vector indicating the presence or absence of a set of attributes. We initialize four identical ResNet-50-based [66] networks with the pre-trained Imagenet [38] weights. Each sub-network takes one of the four images as input - the whole image, and the three body parts images. We remove the fully-connected layer of the ResNet-50 models, and replace with our own fully-connected layer of size 512. As the sigmoid activation function outputs a probability between 0 and 1 (Section 2.3.3), it is ideal for use in attribute prediction, and is therefore used by the final fully-connected layer of our network. To prevent overfitting, we use a dropout of 0.5. The outputs of each sub-network are concatenated to form a 2048-dimensional feature, which is used as our final deep attribute feature. The architecture for our model is shown in Figure 5.3. Examples of attribute detection accuracy on the VIPeR data set can be seen in Table 5.1.

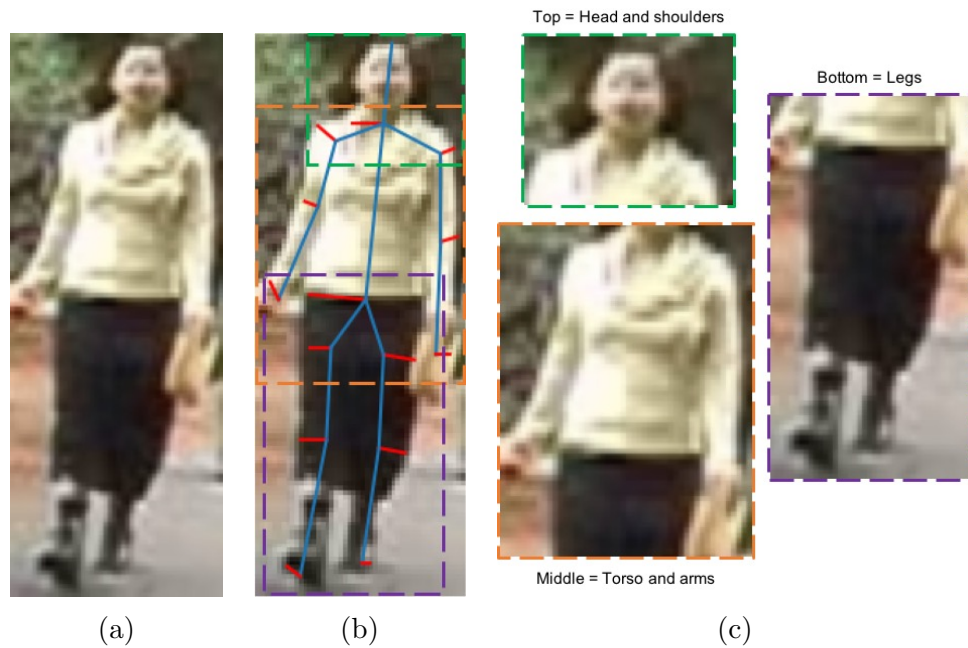


Figure 5.2: An example of how we divide each Re-ID image into three parts-based images: top, middle and bottom, using our deep CNN-based method proposed in Chapter 4. We create a bounding box around each part, and add padding of 15% in the x and y dimensions, to account for any errors in skeleton prediction. We use the original image and the three parts-based images as input to our attribute prediction model. (a) The original input image; (b) The original image with the skeleton and parts separation overlaid; (c) The individual body parts images.

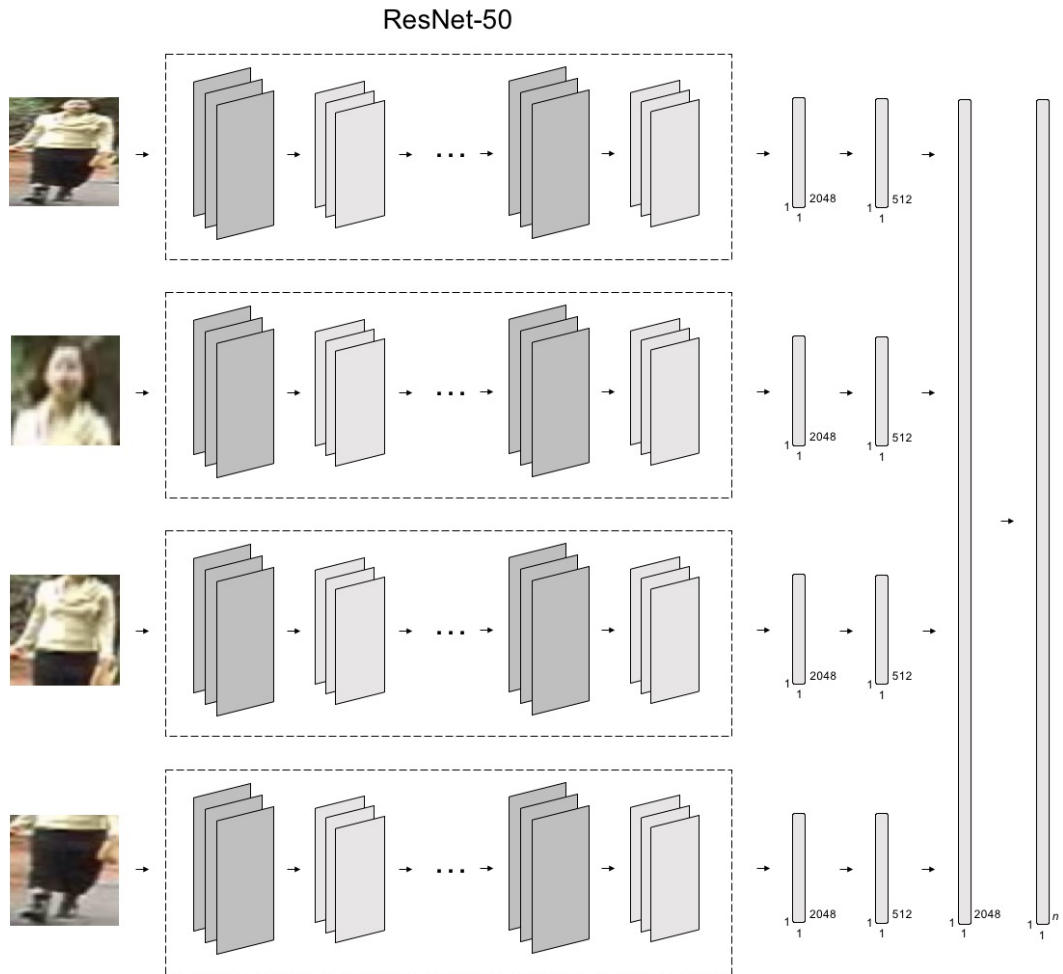


Figure 5.3: The network architecture of the attribute prediction model. The original image is divided into three body parts - the top, middle and bottom. The original image, as well as the three body parts images, are passed through an identical ResNet-50 [66] network architecture. The fully-connected layers of each ResNet-50 model are removed and replaced with our own fully-connected layer of size 512. The four fully-connected layers are then concatenated to form a layer of size 2048. Finally, we append a fully-connected layer of size n , with n being the number of attributes being predicted.

5.3 Results and Discussion

In this section, we will discuss in detail the protocols used when training and testing our method.

Attribute	Accuracy (%)	Attribute	Accuracy (%)
lowerBodySuits	99.8	carryingPlasticBags	96.5
footwearStocking	98.4	lowerBodyCasual	96.4
upperBodyPlaid	97.8	lowerBodyShortSkirt	96.2
upperBodyFormal	97.5	upperBodyRed	95.8
lowerBodyFormal	96.9	upperBodyCasual	95.4

Attribute	Accuracy (%)	Attribute	Accuracy (%)
footwearSneaker	66.5	footwearWhite	60.0
personalLess30	64.5	accessoryNothing	57.2
lowerBodyBlack	63.8	footwearShoes	57.1
lowerBodyGrey	63.7	footwearBlack	55.1
carryingNothing	62.7	upperBodyOther	53.4

Table 5.1: Attribute detection accuracy on the VIPeR [59] data set. The ten best and worst attributes detection accuracies are shown.

5.3.1 Training the Skeleton Prediction Model

In order to demonstrate the generalisation capabilities of our proposed method, we evaluate on four data sets, whilst training on a separate set of data sets. For the skeleton prediction model, we use the 3DPeS [6–8] and QMUL GRID [117, 124, 125] data sets for training. For the 3DPeS data set, we take approximately 80% of all identities as training, and the remaining 20% as validation. For QMUL GRID, similar to in Section 4.3.1, we separate the data set into images which form an image pair, and those which don’t, and take 80% of the identities from each set to contribute to the training set, whilst the remaining 20% contribute to the validation set. This ensures that the 80/20 split relates not just to identities, but also to images. We train the final two fully-connected layers for 15 epochs with a batch size of 32, followed by training the layers from ResNet-50’s third stage onwards, similarly for 15 epochs with a batch size of 32. We utilise the RMSProp [177] optimizer, with a learning rate of 0.001, and use a mean squared error loss.

5.3.2 Training the Attribute Prediction Model

We train the attribute prediction model on a subset of the PETA [39] data set, containing the 3DPeS [6–8], QMUL GRID [117, 124, 125], CAVIAR4REID [29], CUHK [109–111], MIT [142], SARC3D [8] and TownCentre [12–14] data sets. We allocate approximately 80% of the identities from each data set to contribute towards the training set, with the remaining 20% contributing towards the validation set.



Figure 5.4: Examples of attribute prediction accuracy. All images shown in (a) are predicted to be wearing red clothing on their upper body, whilst images in (b) are predicted to be wearing a backpack. Images correctly classified (true-positive) are highlighted in green, whilst those incorrectly classified (false-positive) are highlighted in red. The predicted probability of the presence of each attribute is shown below each image.

The PETA [39] data set provides information for each identity on the presence or absence of 105 attributes, such as the length of their sleeves and whether or not a person is wearing jeans or a backpack. We select the fifty most common attributes, and produce a binary vector denoting the presence or absence of these attributes.

We first train the attribute prediction model from our appended 512-dimensional fully-connected layer onwards, for five epochs using a batch size of 16. We follow by training all layers from ResNet-50’s third-stage onwards, for an additional thirty epochs, maintaining the batch size at 16. We use the Adam [92, 151] optimizer with a learning rate of 10^{-5} , and a binary cross entropy (BCE) loss. Examples of true-positive and false-positive classified images for two attributes can be seen in Figure 5.4.

5.3.3 Evaluation

In this subsection, we will discuss the experimental settings used to evaluate our proposed method. We evaluate on the VIPeR [59], PRID2011 [70], i-LIDS [57, 225, 226] and Market-1501 [219] data sets. Whilst traditional Re-ID methods using attributes generally used the predicted attribute vector as a feature [3, 101, 102], newer methods utilising CNNs often instead extract the penultimate or antepenultimate layer to act as a feature descriptor [129, 171, 174], and have observed greater rank- n scores. In contrast to using the predicted attribute vector, the penultimate or antepenultimate layers are not limited to the typically small dimensionality of the attribute feature, and therefore may be more discriminative [174]. Hence, for each data set, we choose the penultimate, 2048-dimensional layer from our attribute prediction network to be our deep attribute feature descriptor. We apply ℓ_2 -normalization to all feature descriptors prior to matching. We choose to learn a distance metric between feature descriptors using XQDA [115], rather than using Euclidean or cosine distance, in order to compute a distance metric optimal to the Re-ID problem within the attribute context. We use the same hyperparameters for the XQDA distance metric that were used in Chapter 3 and Chapter 4. Whilst we do not use any images from the VIPeR, PRID2011, i-LIDS or Market-1501 data sets to train the skeleton or attribute prediction networks, we do designate a subset of images from these data sets as training images for the XQDA distance metric when testing on these data sets.

Evaluation on the VIPeR data set

We randomly select 316 identities from the VIPeR [59] data set for training the XQDA distance metric, whilst the remaining 316 identities are used for testing. In order to evaluate the effect and contribution of our deep attribute feature descriptor when combined with traditional hand-crafted features, we extract LOMO features [115] from the original images only, and concatenate with our deep attribute features. We test using a single-shot approach, and carry out our experimentation ten times using repeated hold-out validation, averaging to produce the final result. Table 5.2 shows a comparison between our proposed method, which we name *Deep Features & Attribute Detection* (DFAD).

	VIPeR			
	r=1	r=5	r=10	r=20
DFAD (+ LOMO + XQDA)	45.7	76.0	85.2	94.2
DFAD (+ XQDA)	16.3	38.4	52.4	66.6
WSMTAL (+ XQDA) [174]	47.1	71.5	80.3	88.2
BPBPR [209]	44.7	-	84.5	92.1
DLDAFN* [201]	44.1	72.6	81.7	91.5
AFSB (+ LOMO + XQDA) [3]	43.9	-	86.6	94.6
CVSP (+ LOMO)* [34]	43.0	73.0	84.2	92.8
FT-CNN (Comb. + Multi) (+ XQDA) [129]	42.5	72.0	83.0	92.0
MTL-LOREA [170]	42.3	72.2	81.6	89.6
LOMO (+ XQDA)* [115]	38.4	69.4	80.5	91.5
JLAC [90]	29.5	60.3	76.0	87.3
SCAKR (Kernel + Attributes) [40]	28.0	57.1	70.8	83.7
SCAKR (Kernel only)* [40]	26.3	54.7	68.4	81.7
ACSM [120]	16.4	34.3	45.2	-
AFSB (+ XQDA) [3]	13.4	-	72.5	93.3
SCAKR (Attributes only) [40]	10.1	24.4	35.3	48.8

Table 5.2: Results on the VIPeR [59] data set. The best results are highlighted in bold. Results marked with * do not incorporate attribute features.

From Table 5.2, we can see that our attribute model performs strongly against competing state-of-the-art methods. By combining the deep attribute features with the LOMO [115] feature descriptor and applying the XQDA [115] distance metric learning method, we are able to achieve a 7.3% increase when compared to using LOMO (+ XQDA) alone. When using the deep attribute feature alone, our proposed method performs only 0.1% lower than the closest attribute-only method, ACSM [120], in rank-1 score, but exceeds ACSM [120] in the rank-5 and rank-10 scores.

Evaluation on the PRID2011 data set

For PRID2011 [70], we randomly select 100 identities from the 200 identities present within both cameras to become our testing set. The remaining 100 identities act as training data for the XQDA distance metric. As stated in Section 2.7.4, PRID2011 contains an additional 549 identities which are only present in one camera, and as such cannot be used for matching. Therefore, we enlarge the testing gallery set by

also using the additional 549 identities for this purpose. We test using a single-shot approach, and carry out our experimentation ten times using repeated hold-out validation, averaging to produce the final result. Table 5.3 shows the results of our propose method, as well as competing state-of-the-art methods.

	PRID2011			
	r=1	r=5	r=10	r=20
DFAD (+ LOMO + XQDA)	32.9	55.7	67.7	79.4
DFAD (+ XQDA)	13.2	24.8	32.5	45.6
BPBRP [209]	28.2	-	61.0	70.4
WSMTAL (+ XQDA) [174]	24.4	52.3	62.5	74.2
LOMO (+ XQDA)* [115]	24.2	48.2	59.3	71.3
MTL-LOREA [170]	18.0	37.4	50.1	66.6
RF+MA+AC [173]	6.5	22.0	32.5	47.6

Table 5.3: Results on the PRID2011 [70] data set. The best results are highlighted in bold. Results marked with * do not incorporate attribute features.

We can see from the information in Table 5.3 that our proposed method also performs well on the PRID2011 data set. When comparing our proposed method (including LOMO) versus using the LOMO features alone, we observe an increase in the rank-1 score of 8.7%. We can also see that our method achieves significant improvements across all measured rank- n scores.

Evaluation on the i-LIDS data set

For i-LIDS [57, 225, 226], similarly to [40], we divide the data into 69 identities to be used for training the XQDA distance metric, whilst the other 50 identities are used for testing. We test using a single-shot approach, and carry out our experimentation ten times using repeated hold-out validation, averaging to produce the final result. Table 5.4 shows the rank- n scores achieved by both of our proposed approaches, as well as competing state-of-the-art methods.

	i-LIDS			
	r=1	r=5	r=10	r=20
DFAD (+ LOMO + XQDA)	57.3	85.0	92.8	97.4
DFAD (+ XQDA)	45.8	76.4	87.1	94.8
LOMO (+ XQDA)* [115]	48.4	76.4	87.1	95.3
SCAKR (Kernel + Attributes) [40]	44.1	64.9	76.3	89.2
SCAKR (Kernel only)* [40]	42.7	62.0	74.6	86.7
SCAKR (Attributes only) [40]	21.7	41.3	56.8	77.0

Table 5.4: Results on the i-LIDS [57, 225, 226] data set. The best results are highlighted in bold. Results marked with * do not incorporate attribute features.

Table 5.4 shows that our combined attribute and LOMO approach achieves a 8.9% increase in the rank-1 score compared to using the LOMO features alone. However, the proposed attributes-only method also performs significantly better than competing approaches, achieving a 24.1% increase in the rank-1 score versus the closes attribute-only method, SCAKR (Attributes only) [40].

Evaluation on the Market-1501 data set

We additionally experiment on the Market-1501 [219] data set. We base our evaluation on the code provided by [219], and also use the pre-defined training and testing splits, consisting of 750 and 751 identities respectively. We test using a single-shot approach. Table 5.5 shows the rank- n scores achieved by our proposed approaches, as well as competing state-of-the-art methods.

	Market-1501				
	r=1	r=5	r=10	r=20	mAP
DFAD (+ LOMO + XQDA)	53.5	74.8	83.2	88.4	29.8
DFAD (+ XQDA)	20.9	43.6	54.6	66.1	9.8
APR [116]	87.0	95.1	96.4	-	66.9
ACRN [162]	83.6	92.6	95.3	97.0	62.6
TJ-AIDL ^{Duke} [186]	58.2	74.8	81.1	86.5	26.5
WSMTAL [174]	49.5	-	-	-	29.2
LOMO (+ XQDA)* [115]	43.2	66.5	75.7	83.0	22.1
AAIPR [211]	40.3	49.2	58.6	-	20.7

Table 5.5: Results on the Market-1501 [219] data set. The best results are highlighted in bold. Results marked with * do not incorporate attribute features.

From Table 5.5, we observe that on the Market-1501 [219] data set, our proposed DFAD (+ LOMO + XQDA) approach achieves a 10.3% increase versus using just the original LOMO features in combination with XQDA. However, even the DFAD (+ LOMO + XQDA) approach is unable to achieve state-of-the-art results. Unlike other data sets commonly used for Re-ID, the Market-1501 data set is significantly large, with many images per identity. As we train and evaluate on distinct data sets, our attribute prediction network does not benefit during training from the large amount of images present within the Market-1501 data set, which may result in lower rank- n scores during evaluation.

5.3.4 Experimentation with different numbers of parts-based images

We assess the contribution of different combinations of parts-based images to our method. For this purpose, we perform matching using variants of our network trained using a different combination of the original and parts-based images as input. First, we train a network using the original image, as well as the three parts-based images, similar to the experimentation in Section 5.3. Secondly, we train using the three parts-based images. Finally, we train using only the original image. Results of the evaluation on the VIPeR [59] data set can be seen in Table 5.6.

	VIPeR			
	r=1	r=5	r=10	r=20
DFAD (+ LOMO + XQDA) (Original + Three Parts-based images)	45.7	76.0	85.2	94.2
DFAD (+ LOMO + XQDA) (Three Parts-based images only)	45.2	74.0	84.8	94.0
DFAD (+ LOMO + XQDA) (Original images only)	39.8	70.6	83.2	92.4
DFAD (+ XQDA) (Original + Three Parts-based images)	16.3	38.4	52.4	66.6
DFAD (+ XQDA) (Three Parts-based images only)	13.7	33.1	44.7	58.7
DFAD (+ XQDA) (Original images only)	8.8	23.5	35.7	50.1

Table 5.6: Results on the VIPeR [59] data set utilising different combinations of the original and parts-based images. Models are trained with BCE loss. The best results are highlighted in bold.

Table 5.6 shows significant variation in rank- n score when altering the input received by the network. When using attribute features in combination with LOMO features (DFAD (+ LOMO + XQDA)), we observe that the highest rank- n scores are obtained when utilising the original and three parts-based images. However, scores on the network trained using only the three parts-based images are only slightly lower, particularly with regards to the rank-1, rank-10 and rank-20 scores. However, when using only the original image as input, rank- n scores are considerably lower, demonstrating the significance of the effect of using the parts-based images on the rank- n score.

Furthermore, when using attribute features only (DFAD (+ XQDA)), i.e. without the LOMO features, we can see from Table 5.6 that the rank- n results are higher when using the original and three parts-based images than using either the original images only or the three parts-based images only. This demonstrates that both the original and three-parts based images are significant for producing robust attribute features which can achieve high rank- n scores.

However, whilst rank- n scores are improved by increasing the number of parts-based images as input to the attribute network, it is unclear whether this was solely down to the pose-informed design of our network, or the increase in

the number of trainable parameters caused by the increase in number of network branches. Further experimentation may be required to determine the exact reason for the increase in rank- n scores when incorporating additional branches into the network.

5.3.5 Class Imbalance

Background

Given the prevalence of each attribute can vary significantly from attribute-to-attribute, several methods have been proposed which attempt to counteract the negative effects of class imbalance. He and Garcia [65] propose various methods for handling imbalanced data, such as random undersampling, where only the positive examples for each class and an equivalent number of negative examples are used during training. This method was used by [102] to train a model for attribute prediction within the context of Re-ID. He and Garcia [65] also proposed random oversampling, which applies data augmentation to inflate the number of examples from the minority class. However, both random oversampling and undersampling have disadvantages: whilst random undersampling results in a balanced data set, it can remove important examples from the training set, discarding potentially important concepts from being used during the training stage. Similarly, random oversampling, whilst also providing a balanced data set, sources multiple examples of the minority class from the same input, and therefore is at an increased risk of overfitting.

An alternative method to counteract the negative effects of class imbalance is to increase to number of examples from the minority class through generating new data. SMOTE [22] creates synthetic data as an alternative to oversampling, instead generating data which is similar to pre-existing samples from the minority class, instead of simple augmentations. Sharma et al. [163] propose a method of generating synthetic images to enlarge the number of positive samples by taking as context both the visual characteristics of the minority and majority classes. Hence, synthetic data that is the same Mahalanobis distance from the majority classes as the known minority classes can be generated. Furthermore, Generative Adversarial Networks (GANs) have been proposed to better generate artificial imagery. Ponce-López et al. [146] proposes the use of a GAN to increase the number of training samples within a given domain. Wu et al. [202] takes this further by proposing the use of a GAN to generate new images of people whilst incorporating attribute information, and hence is able to generate positive examples of attributes which may be under-represented within the training set.

Cost-Sensitive Learning [42] is an alternative method which can be used, which allows for different penalties for different types of misclassifications. Hence, weights which are directly related to the ratio of positive to negative samples of a given class can be applied to weight the cost in such a way to minimise the negative effects of class imbalance. Elkan [42] apply cost-sensitive learning to a standard Bayesian and decision tree learning method, and experiment by altering the balance of positive and negative training examples, observing little effect on the classifiers performance. Huang et al. [75] use a cost-sensitive learning technique to build a network which predicts certain human characteristic attributes, such as “smiling”, “oval face” and “brown hair”, and validates its effectiveness versus baseline methods. We perform further experimentation on our attributes-based method by incorporating a cost-sensitive learning technique known as Weighted Binary Cross Entropy (WBCE) loss, and evaluate its effectiveness on training an attribute-based network for use in Re-ID.

Weighted Binary Cross Entropy (WBCE) Loss

We propose the use of a Weighted Binary Cross Entropy (WBCE) loss function to counteract the negative effects of class imbalance. Let t_i^j represent the i^{th} attribute of the j^{th} person. For each attribute, i , we calculate the ratio of positive to negative occurrences of an attribute by:

$$pos_i = \frac{1}{p} \sum_{j=0}^{p-1} t_i^j, \quad (5.1)$$

$$neg_i = 1 - pos_i = 1 - \frac{1}{p} \sum_{j=0}^{p-1} t_i^j, \quad (5.2)$$

where p is the number of attribute vectors in the training set. These ratios can be used to calculate a weight, w_i , for each attribute i , which is used to weight the cost of a positive error relative to a negative error, where:

$$w_i = \frac{neg_i}{pos_i}, \quad (5.3)$$

Based on the implementations by Tensorflow [1] and Tensorpack [204], we calculate the WBCE, $loss$, as:

$$loss = (\mathbf{1} - \mathbf{z})\mathbf{r} + \mathbf{m}(\log(\mathbf{1} + \exp(-\text{abs}(\mathbf{r}))) + \max(-\mathbf{r}, 0)), \quad (5.4)$$

This outputs a vector containing the component-wise weighted log losses, where \mathbf{z} is the ground-truth attribute vector, \mathbf{r} is the predicted attribute vector and

\mathbf{m} is equal to $(\mathbf{1} + (\mathbf{w} - \mathbf{1})\mathbf{z})$. As \mathbf{z} is a binary vector representing the presence of absence of a set of I attributes:

$$m_i = \begin{cases} w_i, & \text{if } z_i = 1 \\ 1, & \text{if } z_i = 0 \end{cases} \quad (5.5)$$

Letting n be the number of attributes, the final loss value is calculated by weighting each component-wise loss value by its corresponding positive ratio, and calculating the mean, by:

$$\hat{loss} = \frac{1}{I} \sum_{i=1}^I (loss_i \times pos_i). \quad (5.6)$$

Results and Discussion

We evaluate the performance of the WBCE loss versus the previously used BCE loss by performing evaluation on the VIPeR [59], PRID2011 [70], i-LIDS [57, 225, 226] and Market-1501 [219] data sets. We compare rank- n scores obtained when using WBCE loss and BCE loss. Results can be seen in Table 5.7.

	VIPeR			
	r=1	r=5	r=10	r=20
DFAD (+ LOMO + XQDA) (WBCE)	47.2	76.1	86.7	94.7
DFAD (+ LOMO + XQDA) (BCE)	45.7	76.0	85.2	94.2
DFAD (+ XQDA) (WBCE)	17.0	41.1	54.7	69.0
DFAD (+ XQDA) (BCE)	16.3	38.4	52.4	66.6

Table 5.7: Results on the VIPeR [59] data set utilising WBCE loss and BCE loss. The best results are highlighted in bold.

	PRID2011			
	r=1	r=5	r=10	r=20
DFAD (+ LOMO + XQDA) (WBCE)	32.8	56.4	68.1	78.3
DFAD (+ LOMO + XQDA) (BCE)	32.9	55.7	67.7	79.4
DFAD (+ XQDA) (WBCE)	13.6	28.9	38.8	50.0
DFAD (+ XQDA) (BCE)	13.2	24.8	32.5	45.6

Table 5.8: Results on the PRID2011 [70] data set utilising WBCE loss and BCE loss. The best results are highlighted in bold.

	i-LIDS			
	r=1	r=5	r=10	r=20
DFAD (+ LOMO + XQDA) (WBCE)	58.5	83.9	92.5	97.5
DFAD (+ LOMO + XQDA) (BCE)	57.3	85.0	92.8	97.4
DFAD (+ XQDA) (WBCE)	43.9	77.3	86.9	95.0
DFAD (+ XQDA) (BCE)	45.8	76.4	87.1	94.8

Table 5.9: Results on the i-LIDS [57, 225, 226] data set utilising WBCE loss and BCE loss. The best results are highlighted in bold.

	Market-1501				
	r=1	r=5	r=10	r=20	mAP
DFAD (+ LOMO + XQDA) (WBCE)	53.8	74.7	82.8	88.8	30.0
DFAD (+ LOMO + XQDA) (BCE)	53.5	74.8	83.2	88.4	29.8
DFAD (+ XQDA) (WBCE)	20.4	42.0	53.6	64.8	9.6
DFAD (+ XQDA) (BCE)	20.9	43.6	54.6	66.1	9.8

Table 5.10: Results on the Market-1501 [219] data set utilising WBCE loss and BCE loss. The best results are highlighted in bold.

From Tables 5.7-5.10, we can see that both WBCE loss and BCE loss are able to achieve the highest result at different ranks and on different data sets. However, rank- n results achieved by using the two methods are typically similar. The greatest increase in rank- n score is seen when experimenting on the VIPeR [59] data set, where all rank- n scores when utilising WBCE loss are higher than their corresponding rank- n scores achieved when using BCE loss. However, a significant increase in rank-1 score is also observed on the i-LIDS [57, 225, 226] data set.

Attribute Distribution Analysis

To investigate why these increases are observed, we investigate the distribution of attributes across the data sets used for training and evaluating our attribute model. To calculate the distribution of attributes from each data set, we create a 50-dimensional feature descriptor for each data set consisting of the proportion of positive samples for each attribute, which can be seen in Figure 5.5. We do not perform this evaluation on the Market-1501 [219] data set as the PETA [39] data set does not include ground-truth attribute labelling for this data set.

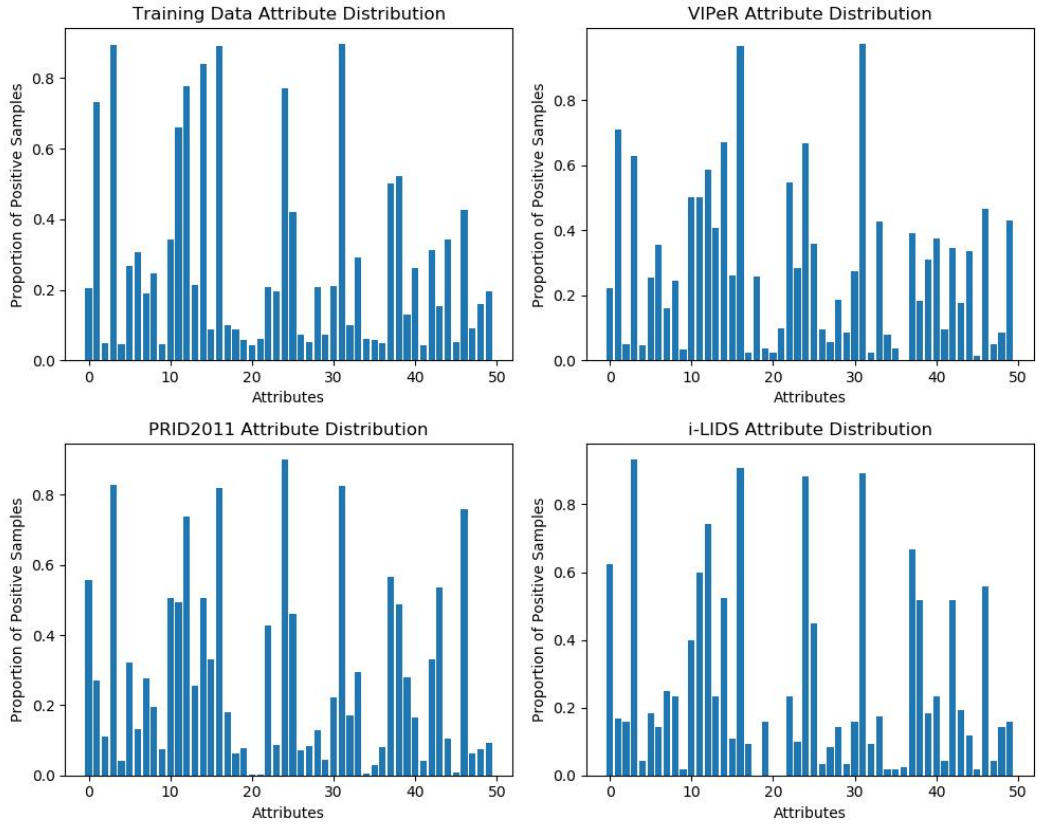


Figure 5.5: The distribution of attributes on the data sets used to train the attribute model, versus the three data sets used to evaluate the attribute model.

In order to compare the distribution of the training data set and the data sets used to evaluate the model, we calculate the cosine distance between the training data set attribute distribution and each of the evaluation data sets attribute distributions. We display the cosine distance between the attribute distributions of the training data set and evaluation data sets in Table 5.11.

VIPeR	PRID2011	i-LIDS
0.049	0.076	0.061

Table 5.11: The cosine distance between the attribute distribution of the training data set and each evaluation data set.

From Table 5.11, we can see the lowest value is achieved on the VIPeR [59] data set, showing that the attribute distribution from the training data set is most similar to the attribute distribution of the VIPeR data set. It can therefore be argued that the similarity in attribute distribution explains the increase in matching performance, seen on the VIPeR data set in Table 5.7 across all rank- n values. The

next lowest distance is observed on the i-LIDS [57, 225, 226] data set, where the rank-1 and rank-20 scores are observed to be higher when using WBCE loss versus using BCE loss. However, rank-5 and rank-10 values are higher when using BCE loss. Finally, we observe the largest distance on the PRID2011 [70] data set, which also suffers from lower rank-1 and rank-20 matching scores, whilst only seeing very small increases in rank-5 and rank-20 score. This investigation serves as a preliminary indicator that data sets with more similar attribute distribution to the training data set are more likely to see an increase when using a weighted loss function. To provide more concrete proof, further research would need to be carried out on greater than three data sets, as a solid conclusion cannot be drawn with such a small number of data points.

5.4 Summary

In this chapter, we have proposed a novel, deep approach to extracting feature descriptors from a network trained using attribute labels. We divide each Re-ID image into three parts: top, middle and bottom, using the deep CNN-based network discussed in Chapter 4. We then pass the original image as well as the three parts images as input to our deep attribute prediction network. The spatial separation of Re-ID images into parts images aids with attribute prediction, given the high correlation between the presence of certain attributes (i.e. *hair length*, *upper body colour* and *wearing shorts*) and specific regions on a persons body. We extracted the penultimate fully-connected layer to use as our deep attribute feature descriptor. We demonstrated that our deep attribute features work not only as a standalone feature descriptor, but can also be used in cooperation with hand-crafted features, in this case the LOMO [115] feature descriptor, in order to improve matching rates even further. Given the imbalance of attributes within the PETA [39] data set, we also propose utilising a Weighted Binary Cross Entropy (WBCE) function which weights the cost of a positive error relative to a negative error, depending on the ratio of positive to negative instances of each attribute, mitigating the negative effects of class imbalance.

We believe that the advantages of our proposed attribute methods are as follows:

- An improvement to attribute detection via spatial separation - Some attributes, such as whether or not someone is wearing a hat, the colour of their shirt, or whether or not they are carrying a backpack, is highly correlated with a specific region of their body/ within the Re-ID image. By performing skeleton prediction and therefore spatial separation prior to attribute prediction, we

can provide additional spatial context to the network to aid with the attribute prediction process.

- Improving feature learning with attributes whilst still using hand-crafted features - Whilst deep features have demonstrated significant performance increases over recent years when compared to hand-crafted features, deep attribute features do not perform as strongly as alternative deep approaches trained using verification or ID classification loss (Section 2.3). However, attribute features possess the advantage over other approaches of being significantly more invariant to changes in pose, brightness and illumination. We have demonstrated that by combining both traditional hand-crafted features with deep attribute features, we are able to achieve an increase in rank- n matching rates.
- Improving attribute prediction using a Weighted Binary Cross Entropy (WBCE) loss function - When training a deep neural network to perform classification, it is unlikely that all classes will contain an equal or close to equal amount of representation within the training set. This is particularly prevalent within the context of attributes, as some attributes are significantly more common than others. We have performed additional experimentation using a WBCE loss function to minimize this issue, by weighting the cost of a positive error relative to a negative error based on the prevalence of the given attribute within the data set.

Chapter 6

Conclusions

6.1 Summary and Discussion

In this thesis, we have proposed several methods for improving Re-ID matching rates, including foreground modelling, feature weighting and attribute detection.

In Chapter 3, we proposed a PLS-based skeleton prediction network. This model was trained by taking HOG appearance features extracted from images as input, and learning a regression between the appearance features and a set of 29 x and y skeleton keypoints. These skeleton keypoints mark specific regions on a person's body, such as head and torso, as well as relative widths of each limb. Thus, given an unseen image, the PLS-based skeleton prediction model can predict the limbs/ foreground regions of an unseen person. This information was used within the Re-ID pipeline to weight features so that the feature descriptors consisted of information predominantly extracted from foreground regions, and thus were more representative of the person rather than the background or the image as a whole. We found that by using the weighted feature descriptors, we were able to increase the Re-ID matching rates.

However, we found that our PLS-based skeleton prediction model did not generalise well when presented with Re-ID images containing people standing at significantly different orientations relative to the camera. Therefore, this issue required the use of a further PLS-based model, trained to predict the orientation of a given unseen Re-ID image. Then, a separate PLS-based skeleton prediction model was trained for each orientation. In Chapter 4, we overcame this issue by instead replacing the PLS-based skeleton prediction models with a deep CNN-based skeleton prediction network. Unlike the PLS-based models, the deep CNN-based method was able to predict the skeleton of a person within an unseen Re-ID image to a high level of accuracy without the need to train multiple skeleton prediction models. For

the deep CNN-based network, we pass the images and corresponding skeletons for a given training set to the network, which learns a regression between the two. The unseen, testing images are passed to the network, which then outputs a predicted skeleton. This skeleton is used in the same way as in Chapter 3, where features are extracted which are weighted towards the foreground areas. Finally, we carry out distance metric learning and matching.

Moreover, we experimented with utilising deep features; features learnt through supervised learning, rather than manually-specified, hand-crafted features such as colour and texture. Whilst colour and texture are generally discriminative features, they suffer significantly from variation in illumination and pose. In Chapter 5, we obtained our deep features by training a deep CNN-based network which detects the presence of a series of attributes, such as *red shirt* and *short hair*. The penultimate layer of our deep attribute detection network is used as a deep feature, and is also combined with hand-crafted features to improve the matching rates further. We found that our attributes-only and attributes with hand-crafted features approaches performed competitively against other similar approaches. In addition, to counteract the problem of some attributes being much more prevalent within the training data set, we also proposed a Weighted Binary Cross Entropy (WBCE) loss function, which weights the cost of a positive error relative to a negative error during training by the ratio of negative instances to positive instances of a given attribute in the training set. We found that rank- n results were higher when the attribute distribution of the evaluation data set was more similar to the attribute distribution of the training data set.

We evaluated our proposed methods on a selection of standard Re-ID data sets. We found that our methods increased the rank- n matching rates, which measures the cumulative amount of probe images correctly matched to the corresponding gallery image within n guesses. As well as evaluating rank- n matching performance, we also used RMSE to evaluate the performance of our skeleton prediction models. In addition, we used accuracy as a metric to measure the performance of our attribute detection model.

We believe that the strengths of our work are as follows:

1. We have created two skeleton prediction approaches (Chapter 3 and Chapter 4), which were both able to predict highly accurate skeletons when given a unseen image. These skeletons were then used to divide a Re-ID image into the foreground (i.e. the person) and the background regions.
2. We have proposed an extension to the LOMO [115] feature descriptor, Weighted LOMO (Chapter 3), which easily integrates with our predicted skeletons to

provide a weighted representation of a Re-ID image. Evaluation using our extended feature descriptor demonstrated that higher rank- n scores can be achieved when compared to utilising the vanilla LOMO feature descriptor.

3. We have demonstrated that by utilising a deep CNN, we are able to produce a skeleton prediction model which generalises well to a variety of different poses (Chapter 4). This is in comparison to the PLS-based skeleton prediction model 3, which required multiple skeleton prediction models to be trained in order to handle significantly different poses.
4. We have proven that it is possible to create skeleton prediction and attribute prediction models which generalise well between data sets. For example, we were able to utilise our skeleton prediction models (Chapter 3 and Chapter 4) to predict skeletons and perform Re-ID matching on the CUHK03 [111] data set, resulting in increased rank- n scores, even though this data set contributed no training data to the skeleton prediction model. Similarly, we performed evaluation of our attribute prediction model on data sets which were not used to train the model, and still achieved significant increases in rank- n scores.
5. We have performed research which acts as a preliminary indicator that rank- n scores can be improved by the use of a pose-informed multi-branch attribute prediction network, which can then be improved even further by utilising a weighted loss function (Chapter 5).

However, we believe that the weaknesses of our work are as follows:

1. Whilst we have demonstrated increased rank- n matching rates on standard Re-ID data sets, many of these data sets are not very representative of a real-world scenario. For example, the VIPeR [59] data set contains very few examples of occlusion, something which would be very prevalent in the real world. Our work therefore has a limitation in that it may not generalise well to a real-world scenario.
2. Whilst we have demonstrated that our proposed deep skeleton predictor (Chapter 4) produces accurate skeletons, which often lead to better rank- n scores when compared to our PLS-based skeleton predictor (Chapter 3), the experimental settings for these two methods were not identical. This is because the deep skeleton predictor had access to a greater amount of training data.
3. Not all of our hypotheses are fully proven within this work. For example, our work in Chapter 5 serves as a preliminary indicator that evaluation data sets with a similar attribute distribution to the training data set lead to greater

rank- n scores. However, this evaluation was carried out on only three data sets, and hence we cannot draw a solid conclusion from such a small sample.

4. Some of our work could be improved by the inclusion of deep features. For example, in Chapter 3, Chapter 4 and Chapter 5, we utilise the LOMO [115] hand-crafted appearance features, whereas deep features may have produced greater rank- n scores.
5. Whilst we have demonstrated high rank- n scores when evaluating on still images, we have not evaluated our methods on video sequences. Using video sequences may provide a greater insight into the performance of our methods, and be more representative of the problem of Re-ID in a real-world scenario.

6.2 Future Work

We propose the following suggestions for future directions for this research, both to mitigate the limitations of this research and to achieve greater rank- n scores:

1. Additional training data for the skeleton prediction algorithm - Re-ID data sets tend to be small in both overall size, as well as the number of images of each identity. As such, the accuracy of the skeleton prediction models could be significantly improved with a larger quantity of training samples. Specifically, this may have the most impact on the models ability to predict the skeletons of individuals within Re-ID images with more unusual poses, where there may be little to no similar poses within the existing training set.
2. Incorporating occlusion information into the skeleton prediction pipeline - The existing skeleton prediction frameworks which we propose assume all limbs are visible, and not occluded. Consequently, feature descriptors extracted from any given limb could look very different if the limb is occluded, and hence possesses significant differences in visual characteristics. Future work could research alternative approaches to deal with missing information caused by occlusion, including predicting the appearance of a limb given the appearance of another limb.
3. Experimenting with the number of components used in the PLS models - In the research presented in Chapter 3, we used the top 15 principal components extracted from each image during the skeleton prediction stage, and the top 50 principal components during the orientation prediction stage. Further research could involve determining the most accurate number of principal components to achieve the best skeleton and orientation prediction.

4. Replacing the PLS regression-based orientation group prediction model with a classification method - To predict an orientation group as part of the work presented in Chapter 3, we use a PLS regression-based model, which learns a mapping between input HOG appearance features X , and an orientation group label $Y \in \{0, \dots, n - 1\}$, where n is the number of distinct orientation groups. Whilst we observed good accuracy by utilising this method, accuracy could be increased further by utilising a classification model, such as SVM [31] or Naïve Bayes [128, 134].
5. Improving the use of foreground modelling during the limb-by-limb level SCNCD feature descriptor extraction - Currently, limb-by-limb level SCNCD feature descriptors extracted as part of the work proposed in Chapter 3 and Chapter 4 are extracted from a defined bounding box without taking into account potential negative effects caused by poor skeleton prediction. This is in comparison to the proposed Weighted LOMO approach, which minimises the negative impact of poor skeleton prediction by instead weighting image patches according to the percentage of foreground within the patch. Future work could look into how to similarly weight the limb-by-limb level SCNCD features in a manner which mitigates the negative impact of poor skeleton prediction.
6. Experimentation with speeding up training: Incorporating deep learning leads to a longer training time versus more shallow methods. Further experimentation could be carried out to work to decrease training time, ensuring the approach is scalable to larger data sets.
7. Experimentation with using higher resolution feature maps: The later layers of the ResNet-50 model used by DNAM are low resolution, with the final convolutional layer outputting feature maps of only 7×7 pixels. More accurate skeleton prediction results may be achieved if higher resolution feature maps are utilised, as accurate positions of individual limbs may not be obtainable at such low resolutions.
8. A more standardised comparison between the PLS-based and deep skeleton prediction approaches - Whilst the deep skeleton prediction approach discussed in Chapter 4 demonstrated positive results when compared to the PLS-based approach, it did also benefit from a larger data set. Further experimentation could compare the two approaches using an equivalent training set, to provide a more thorough comparison between the two.
9. Further experimentation with the choice of model used for skeleton prediction

- In Chapter 3, we utilised PLS regression for skeleton prediction, whereas in Chapter 4, we instead utilised a CNN. Further experimentation could investigate alternative methods such as canonical correlation analysis (CCA) or a multilayer perceptron.
10. Further experimentation on the benefit of multiple network branches in the Deep Features & Attribute Detection (DFAD) network - We presented a four-branch attribute prediction network, with each branch taking as input a different region of a given Re-ID image. The four image regions were the original image, and three parts-based images determined by a skeleton prediction model. We observed higher rank- n scores when using a higher number of inputs images, however, it is unclear whether this was down to the pose-informed design of our method, or the increase in the number of trainable parameters when using a higher number of network branches. Future work could investigate this uncertainty to determine the whole reason for the increase in rank- n scores.
 11. More concrete evaluation of the Weighted Binary Cross Entropy (WBCE) loss function using more than three data sets - In Chapter 5, it was discussed that evaluation data sets with a more similar attribute distribution to the training data set demonstrated a greater increase in rank- n scores when using a Weighted Binary Cross Entropy (WBCE) loss function. However, this relation could not be concretely determined using only three data sets. Further work could attempt to prove or disprove this hypothesis by experimenting on a greater number of data sets.
 12. Greater experimentation on the benefit of using the penultimate layer of the network, rather than the final layer, as the deep attribute feature descriptor in the Deep Features & Attribute Detection (DFAD) network - Following similar literature [129, 171, 174], we choose a high-dimensional fully-connected layer from near the output of our network, rather than the predicted attribute vector, to use as a deep attribute feature descriptor. Further experimentation could determine whether or not this leads to an increase in rank- n score, as well as investigate which layer of the network produces the highest rank- n scores.
 13. Evaluation of the worth of supervised attribute features over supervised ID features - To build and train an attribute prediction network, the user must gather a set of labelled attribute data, and follow by designing an appropriate network architecture. However, attribute features alone have not been able to produce state-of-the-art results, instead often being combined with other approaches, including hand-crafted features, to increase the overall rank- n

scores. For the same amount of work, a user could instead design and build a network trained on ID-labelled data, which has been shown to produce high rank- n scores [108, 112, 203, 206]. Whilst our work demonstrates that pose-informed deep attribute features can be combined with hand-crafted features to produce high rank- n scores, it does not demonstrate whether or not pose-informed deep attribute features can work alongside the state-of-the-art deep features which are currently obtaining the highest rank- n scores. Future work could investigate the worth of pose-informed deep attribute features in comparison to deep features obtained through a network trained using ID Classification loss, including a potential concatenation between the two to create a more robust deep feature descriptor.

14. Using Generative Adversarial Networks (GANs) [54, 228] - Recent work in Re-ID [119, 149, 192, 228, 231] are utilising Generative Adversarial Networks (GANs) within the Re-ID framework. For example, GANs are being used to generate additional training data [119, 228], or to take a Re-ID image and generate a new image of the same individual in an alternate pose [119, 149]. Additional training data, especially if the generated data contains images of those with unusual poses, could significantly benefit the quality of the skeleton prediction network's output. Furthermore, using a GAN to standardise the pose of a set of individuals can minimise the negative effects of pose variation. Within the context of attribute detection networks, a GAN could be also used to generate positive samples which contain the presence of attributes which are otherwise rare within the training set, reducing the negative effects of class imbalance during training.
15. Applying the novel approaches proposed within this thesis to Re-ID video sequences - Whilst our proposals have been applied to individual Re-ID images, we have not utilised video sequences. However, video sequences could be incorporated using techniques such as Long Short-Term Memory networks (LSTM) [71] to process a series of data. Using this method, information extracted from the first frame in a sequence can be propagated all the way to the final frame in the sequence, allowing a sequence-level feature descriptor to be obtained through extracting the most discriminative features from the sequence as a whole.

Appendix A

Data Annotation

Due to the absence of foreground information for the common Re-ID data sets, we manually labelled the VIPeR [59], QMUL GRID [117, 125] and 3DPeS [7, 8] data sets with skeletal information. Each skeleton contains twenty-nine keypoints which represent fourteen limbs, with each limb consisting of two end-points and a third point representing the edge of the limb. The bottom keypoint of each limb also acts as the top keypoint of the following limb. Figure A.1 shows examples of images and their corresponding hand-labelled skeletons. We created a MATLAB Graphical User Interface (GUI) to aid the labelling process. An example of the labelling GUI can be seen in Figure A.2.



Figure A.1: Example images and their corresponding hand-labelled skeletons from the VIPeR [59] data set.



Figure A.2: An example of the skeleton labelling GUI using images from the VIPeR [59] data set. The first Re-ID image represents the input image on which the user clicks to mark skeleton keypoints. The second Re-ID image shows the recorded skeleton keypoints converted to a skeleton and overlaid on the input image in real-time. The third image is a static reference image showing the order in which the skeleton keypoint should be collected. Finally, some information on the Re-ID image is shown on the right of the GUI.

Bibliography

- [1] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. URL <https://www.tensorflow.org/>. Software available from tensorflow.org.
- [2] Ejaz Ahmed, Michael Jones, and Tim K Marks. An improved deep learning architecture for person re-identification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3908–3916, 2015.
- [3] Le An, Xiaojing Chen, Shuang Liu, Yinjie Lei, and Songfan Yang. Integrating appearance features and soft biometrics for person re-identification. *Multimedia Tools and Applications*, 76(9):12117–12131, 2017.
- [4] Mykhaylo Andriluka, Stefan Roth, and Bernt Schiele. Pictorial structures revisited: People detection and articulated pose estimation. In *2009 IEEE conference on computer vision and pattern recognition*, pages 1014–1021. IEEE, 2009.
- [5] Alexandru O Balan, Michael J Black, Horst Haussecker, and Leonid Sigal. Shining a light on human pose: On shadows, shading and the estimation of pose and shape. In *2007 IEEE 11th International Conference on Computer Vision*, pages 1–8. IEEE, 2007.
- [6] Davide Baltieri, Roberto Vezzani, and Rita Cucchiara. 3d body model con-

- struction and matching for real time people re-identification. In *Eurographics Italian Chapter Conference*, pages 65–71, 2010.
- [7] Davide Baltieri, Roberto Vezzani, and Rita Cucchiara. 3dpes: 3d people dataset for surveillance and forensics. In *Proceedings of the 2011 joint ACM workshop on Human gesture and behavior understanding*, pages 59–64. ACM, 2011.
- [8] Davide Baltieri, Roberto Vezzani, and Rita Cucchiara. Sarc3d: a new 3d body model for people tracking and re-identification. In *International conference on image analysis and processing*, pages 197–206. Springer, 2011.
- [9] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. In *European conference on computer vision*, pages 404–417. Springer, 2006.
- [10] Amy Bearman and Catherine Dong. Human pose estimation and activity classification using convolutional neural networks. *CS231n Course Project Reports*, 2015.
- [11] Rodrigo Benenson, Mohamed Omran, Jan Hosang, and Bernt Schiele. Ten years of pedestrian detection, what have we learned? In *European Conference on Computer Vision*, pages 613–627. Springer, 2014.
- [12] Ben Benfold and Ian Reid. Stable multi-target tracking in real-time surveillance video. In *CVPR 2011*, pages 3457–3464. IEEE, 2011.
- [13] Ben Benfold and Ian D Reid. Colour invariant head pose classification in low resolution video. In *BMVC*, pages 1–10, 2008.
- [14] Ben Benfold and Ian D Reid. Guiding visual surveillance by tracking human attention. In *BMVC*, volume 2, number 6, page 7, 2009.
- [15] Yoshua Bengio, Patrice Simard, Paolo Frasconi, et al. Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 5(2):157–166, 1994.
- [16] John S Bridle. Training stochastic model recognition algorithms as networks can lead to maximum mutual information estimation of parameters. In *Advances in neural information processing systems*, pages 211–217, 1990.
- [17] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 132–149, 2018.

- [18] Rich Caruana. Multitask learning: A knowledge-based source of inductive bias. In *ICML*, 1993.
- [19] Mahalanobis Prasanta Chandra et al. On the generalised distance in statistics. In *Proceedings of the National Institute of Sciences of India*, volume 2, number 1, pages 49–55, 1936.
- [20] Chih-Chung Chang and Chih-Jen Lin. Libsvm: A library for support vector machines. *ACM transactions on intelligent systems and technology (TIST)*, 2(3):27, 2011.
- [21] Xiaobin Chang, Timothy M Hospedales, and Tao Xiang. Multi-level factorisation net for person re-identification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2109–2118, 2018.
- [22] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357, 2002.
- [23] Dapeng Chen, Zejian Yuan, Gang Hua, Nanning Zheng, and Jingdong Wang. Similarity learning on an explicit polynomial kernel feature map for person re-identification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1565–1573, 2015.
- [24] Dapeng Chen, Zejian Yuan, Badong Chen, and Nanning Zheng. Similarity learning with spatial constraints for person re-identification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1268–1277, 2016.
- [25] Huizhong Chen, Andrew Gallagher, and Bernd Girod. Describing clothing by semantic attributes. In *European conference on computer vision*, pages 609–623. Springer, 2012.
- [26] Jiaxin Chen, Yunhong Wang, Jie Qin, Li Liu, and Ling Shao. Fast person re-identification via cross-camera semantic binary transformation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3873–3882, 2017.
- [27] Weihua Chen, Xiaotang Chen, Jianguo Zhang, and Kaiqi Huang. A multi-task deep network for person re-identification. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

- [28] De Cheng, Yihong Gong, Sanping Zhou, Jinjun Wang, and Nanning Zheng. Person re-identification by multi-channel parts-based cnn with improved triplet loss function. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1335–1344, 2016.
- [29] Dong Seon Cheng, Marco Cristani, Michele Stoppa, Loris Bazzani, and Vittorio Murino. Custom pictorial structures for re-identification. In *Bmvc*, volume 1, number 2, page 6. Citeseer, 2011.
- [30] Marc Claesen and Bart De Moor. Hyperparameter search in machine learning. *arXiv preprint arXiv:1502.02127*, 2015.
- [31] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.
- [32] Stanford CS231n. Cs231n: Convolutional neural networks for visual recognition. URL: <https://cs231n.github.io/neural-networks-1/> (visited on 07/02/2019), 2016.
- [33] Stanford CS231n. Cs231n: Convolutional neural networks for visual recognition. URL: <http://cs231n.github.io/convolutional-networks/> (visited on 12/12/2019), 2016.
- [34] Ju Dai, Ying Zhang, Huchuan Lu, and Hongyu Wang. Cross-view semantic projection learning for person re-identification. *Pattern Recognition*, 75:63–76, 2018.
- [35] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *international Conference on computer vision & Pattern Recognition (CVPR'05)*, volume 1, pages 886–893. IEEE Computer Society, 2005.
- [36] Jason V Davis, Brian Kulis, Prateek Jain, Suvrit Sra, and Inderjit S Dhillon. Information-theoretic metric learning. In *Proceedings of the 24th international conference on Machine learning*, pages 209–216. ACM, 2007.
- [37] Sijmen De Jong. Simpls: an alternative approach to partial least squares regression. *Chemometrics and intelligent laboratory systems*, 18(3):251–263, 1993.
- [38] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

- [39] Yubin Deng, Ping Luo, Chen Change Loy, and Xiaoou Tang. Pedestrian attribute recognition at far distance. In *Proceedings of the 22nd ACM international conference on Multimedia*, pages 789–792. ACM, 2014.
- [40] Husheng Dong, Chunping Liu, Yi Ji, Zhaohui Wang, and Shengrong Gong. Fusion of spatially constrained attributes with kernelized ranking for person re-identification. In *2015 12th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 1–6. IEEE, 2015.
- [41] Carsten F Dormann, Jane Elith, Sven Bacher, Carsten Buchmann, Gudrun Carl, Gabriel Carré, Jaime R García Marquéz, Bernd Gruber, Bruno Lafourcade, Pedro J Leitão, et al. Collinearity: a review of methods to deal with it and a simulation study evaluating their performance. *Ecography*, 36(1):27–46, 2013.
- [42] Charles Elkan. The foundations of cost-sensitive learning. In *International joint conference on artificial intelligence*, volume 17, number 1, pages 973–978. Lawrence Erlbaum Associates Ltd, 2001.
- [43] Michela Farenzena, Loris Bazzani, Alessandro Perina, Vittorio Murino, and Marco Cristani. Person re-identification by symmetry-driven accumulation of local features. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 2360–2367. IEEE, 2010.
- [44] Pedro F Felzenszwalb, Ross B Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part-based models. *IEEE transactions on pattern analysis and machine intelligence*, 32(9):1627–1645, 2010.
- [45] Itzhak Fogel and Dov Sagi. Gabor filters as texture discriminator. *Biological cybernetics*, 61(2):103–113, 1989.
- [46] Per-Erik Forssén. Maximally stable colour regions for recognition and matching. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2007.
- [47] William T Freeman and Michal Roth. Orientation histograms for hand gesture recognition. In *International workshop on automatic face and gesture recognition*, volume 12, pages 296–301, 1995.
- [48] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. *The elements of statistical learning*, volume 1, number 10. Springer series in statistics New York, 2001.

- [49] Kunihiko Fukushima. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological cybernetics*, 36(4):193–202, 1980.
- [50] Yarín Gal and Zoubin Ghahramani. A theoretically grounded application of dropout in recurrent neural networks. In *Advances in neural information processing systems*, pages 1019–1027, 2016.
- [51] Mengyue Geng, Yaowei Wang, Tao Xiang, and Yonghong Tian. Deep transfer learning for person re-identification. *arXiv preprint arXiv:1611.05244*, 2016.
- [52] Theo Gevers and Arnold WM Smeulders. Color-based object recognition. *Pattern recognition*, 32(3):453–464, 1999.
- [53] Shaogang Gong, Marco Cristani, Chen Change Loy, and Timothy M Hospedales. The re-identification challenge. In *Person re-identification*, pages 1–20. Springer, 2014.
- [54] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.
- [55] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. <http://www.deeplearningbook.org>.
- [56] Mengran Gou, Srikrishna Karanam, Wenqian Liu, Octavia Camps, and Richard J. Radke. Dukemtmc4reid: A large-scale multi-camera person re-identification dataset. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, July 2017.
- [57] Gov.uk. Imagery library for intelligent detection systems. <https://www.gov.uk/guidance/imagery-library-for-intelligent-detection-systems>. Accessed: 2019-04-01.
- [58] Douglas Gray and Hai Tao. Viewpoint invariant pedestrian recognition with an ensemble of localized features. In *European conference on computer vision*, pages 262–275. Springer, 2008.
- [59] Douglas Gray, Shane Brennan, and Hai Tao. Evaluating appearance models for recognition, reacquisition, and tracking. In *Proc. IEEE International Workshop on Performance Evaluation for Tracking and Surveillance (PETS)*, volume 3, number 5, pages 1–7. Citeseer, 2007.

- [60] Thomas L Griffiths and Zoubin Ghahramani. The indian buffet process: An introduction and review. *Journal of Machine Learning Research*, 12(Apr): 1185–1224, 2011.
- [61] Richard F Gunst and Robert L Mason. Some considerations in the evaluation of alternate prediction equations. *Technometrics*, 21(1):55–63, 1979.
- [62] Tiansheng Guo, Dongfei Wang, Zhuqing Jiang, Aidong Men, and Yun Zhou. An enhanced deep convolutional neural network for person re-identification. In *2018 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*, pages 1–6. IEEE, 2018.
- [63] Omar Hamdoun, Fabien Moutarde, Bogdan Stanciulescu, and Bruno Steux. Person re-identification in multi-camera system by signature based on interest point descriptors collected on short video sequences. In *2008 Second ACM/IEEE International Conference on Distributed Smart Cameras*, pages 1–6. IEEE, 2008.
- [64] Zellig S Harris. Distributional structure. *Word*, 10(2-3):146–162, 1954.
- [65] Haibo He and Edwardo A Garcia. Learning from imbalanced data. *IEEE Transactions on Knowledge & Data Engineering*, 21(9):1263–1284, 2008.
- [66] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.
- [67] NE Helwig. Multivariate linear regression, 2017.
- [68] Alexander Hermans, Lucas Beyer, and Bastian Leibe. In defense of the triplet loss for person re-identification. *arXiv preprint arXiv:1703.07737*, 2017.
- [69] Geoffrey E Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*, 2012.
- [70] Martin Hirzer, Csaba Beleznai, Peter M Roth, and Horst Bischof. Person re-identification by descriptive and discriminative classification. In *Scandinavian conference on Image analysis*, pages 91–102. Springer, 2011.
- [71] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

- [72] Alan L Hodgkin and Andrew F Huxley. A quantitative description of membrane current and its application to conduction and excitation in nerve. *The Journal of physiology*, 117(4):500–544, 1952.
- [73] HAROLD HOTELLING. Relations between two sets of variates. *Biometrika*, 28(3-4):321–377, 1936.
- [74] Gang Hua, Ming-Hsuan Yang, and Ying Wu. Learning to estimate human pose with data driven belief propagation. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 2, pages 747–754. IEEE, 2005.
- [75] Chen Huang, Yining Li, Chen Change Loy, and Xiaoou Tang. Learning deep representation for imbalanced classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5375–5384, 2016.
- [76] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.
- [77] Yan Huang, Hao Sheng, Yanwei Zheng, and Zhang Xiong. Deepdiff: Learning deep difference features on human body parts for person re-identification. *Neurocomputing*, 241:191–203, 2017.
- [78] David H Hubel and Torsten N Wiesel. Receptive fields of single neurones in the cat’s striate cortex. *The Journal of physiology*, 148(3):574–591, 1959.
- [79] David H Hubel and Torsten N Wiesel. Receptive fields, binocular interaction and functional architecture in the cat’s visual cortex. *The Journal of physiology*, 160(1):106–154, 1962.
- [80] David H Hubel and Torsten N Wiesel. Receptive fields and functional architecture of monkey striate cortex. *The Journal of physiology*, 195(1):215–243, 1968.
- [81] Minyoung Huh, Pulkit Agrawal, and Alexei A Efros. What makes imagenet good for transfer learning? *arXiv preprint arXiv:1608.08614*, 2016.
- [82] Robert A Jacobs, Michael I Jordan, Steven J Nowlan, Geoffrey E Hinton, et al. Adaptive mixtures of local experts. *Neural computation*, 3(1):79–87, 1991.
- [83] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. In *Advances in neural information processing systems*, pages 2017–2025, 2015.

- [84] Daniel J Jobson, Zia-ur Rahman, and Glenn A Woodell. A multiscale retinex for bridging the gap between color images and the human observation of scenes. *IEEE Transactions on Image processing*, 6(7):965–976, 1997.
- [85] Daniel J Jobson, Zia-ur Rahman, and Glenn A Woodell. Properties and performance of a center/surround retinex. *IEEE transactions on image processing*, 6(3):451–462, 1997.
- [86] Nebojsa Jojic, Alessandro Perina, Marco Cristani, Vittorio Murino, and Brendan Frey. Stel component analysis: Modeling spatial correlations in image class structure. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2044–2051. IEEE, 2009.
- [87] Ian T Jolliffe. A note on the use of principal components in regression. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 31(3):300–303, 1982.
- [88] Kai Jüngling, Christoph Bodensteiner, and Michael Arens. Person re-identification in multi-camera networks. In *CVPR 2011 WORKSHOPS*, pages 55–61. IEEE, 2011.
- [89] Sham M Kakade and Dean P Foster. Multi-view regression via canonical correlation analysis. In *International Conference on Computational Learning Theory*, pages 82–96. Springer, 2007.
- [90] Sameh Khamis, Cheng-Hao Kuo, Vivek K Singh, Vinay D Shet, and Larry S Davis. Joint learning for attribute-consistent person re-identification. In *European Conference on Computer Vision*, pages 134–146. Springer, 2014.
- [91] Ji-Hyun Kim. Estimating classification error rate: Repeated cross-validation, repeated hold-out and bootstrap. *Computational statistics & data analysis*, 53(11):3735–3745, 2009.
- [92] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [93] Martin Koestinger, Martin Hirzer, Paul Wohlhart, Peter M Roth, and Horst Bischof. Large scale metric learning from equivalence constraints. In *2012 IEEE conference on computer vision and pattern recognition*, pages 2288–2295. IEEE, 2012.
- [94] Ron Kohavi et al. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Ijcai*, volume 14, number 2, pages 1137–1145. Montreal, Canada, 1995.

- [95] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.
- [96] Thomas Kurbiel and Shahrzad Khaleghian. Training of deep neural networks based on distance measures using rmsprop. *arXiv preprint arXiv:1708.01911*, 2017.
- [97] Xiangyang Lan and Daniel P Huttenlocher. Beyond trees: Common-factor models for 2d human pose recovery. In *Tenth IEEE International Conference on Computer Vision (ICCV’05) Volume 1*, volume 1, pages 470–477. IEEE, 2005.
- [98] Edwin H Land and John J McCann. Lightness and retinex theory. *Josa*, 61(1):1–11, 1971.
- [99] Steve Lawrence and C Lee Giles. Overfitting and neural networks: conjugate gradient and backpropagation. In *Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks. IJCNN 2000. Neural Computing: New Challenges and Perspectives for the New Millennium*, volume 1, pages 114–119. IEEE, 2000.
- [100] Ryan Layne, Timothy M Hospedales, and Shaogang Gong. Towards person identification and re-identification with attributes. In *European Conference on Computer Vision*, pages 402–412. Springer, 2012.
- [101] Ryan Layne, Timothy M Hospedales, Shaogang Gong, and Q Mary. Person re-identification by attributes. In *Bmvc*, volume 2, number 3, page 8, 2012.
- [102] Ryan Layne, Timothy M Hospedales, and Shaogang Gong. Attributes-based re-identification. In *Person Re-Identification*, pages 93–117. Springer, 2014.
- [103] Quoc V Le, Marc’Aurelio Ranzato, Rajat Monga, Matthieu Devin, Kai Chen, Greg S Corrado, Jeff Dean, and Andrew Y Ng. Building high-level features using large scale unsupervised learning. *arXiv preprint arXiv:1112.6209*, 2011.
- [104] Laura Leal-Taixé, Anton Milan, Ian Reid, Stefan Roth, and Konrad Schindler. Motchallenge 2015: Towards a benchmark for multi-target tracking. *arXiv preprint arXiv:1504.01942*, 2015.
- [105] Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989.

- [106] Yann LeCun, Léon Bottou, Yoshua Bengio, Patrick Haffner, et al. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86 (11):2278–2324, 1998.
- [107] Bastian Leibe, Aleš Leonardis, and Bernt Schiele. Robust object detection with interleaved categorization and segmentation. *International journal of computer vision*, 77(1-3):259–289, 2008.
- [108] Dangwei Li, Xiaotang Chen, Zhang Zhang, and Kaiqi Huang. Learning deep context-aware features over body and latent parts for person re-identification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 384–393, 2017.
- [109] Wei Li and Xiaogang Wang. Locally aligned feature transforms across views. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3594–3601, 2013.
- [110] Wei Li, Rui Zhao, and Xiaogang Wang. Human reidentification with transferred metric learning. In *Asian Conference on Computer Vision*, pages 31–44. Springer, 2012.
- [111] Wei Li, Rui Zhao, Tong Xiao, and Xiaogang Wang. Deepreid: Deep filter pairing neural network for person re-identification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 152–159, 2014.
- [112] Wei Li, Xiatian Zhu, and Shaogang Gong. Person re-identification by deep joint learning of multi-loss classification. *arXiv preprint arXiv:1705.04724*, 2017.
- [113] Shengcai Liao and Stan Z Li. Efficient psd constrained asymmetric metric learning for person re-identification. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3685–3693, 2015.
- [114] Shengcai Liao, Guoying Zhao, Vili Kellokumpu, Matti Pietikäinen, and Stan Z Li. Modeling pixel process with scale invariant local patterns for background subtraction in complex scenes. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1301–1306. IEEE, 2010.
- [115] Shengcai Liao, Yang Hu, Xiangyu Zhu, and Stan Z Li. Person re-identification by local maximal occurrence representation and metric learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2197–2206, 2015.

- [116] Yutian Lin, Liang Zheng, Zhedong Zheng, Yu Wu, Zhilan Hu, Chenggang Yan, and Yi Yang. Improving person re-identification by attribute and identity learning. *Pattern Recognition*, 2019.
- [117] Chunxiao Liu, Shaogang Gong, Chen Change Loy, and Xinggang Lin. Person re-identification: What features are important? In *European Conference on Computer Vision*, pages 391–401. Springer, 2012.
- [118] Hao Liu, Jiashi Feng, Meibin Qi, Jianguo Jiang, and Shuicheng Yan. End-to-end comparative attention networks for person re-identification. *IEEE Transactions on Image Processing*, 26(7):3492–3506, 2017.
- [119] Jinxian Liu, Bingbing Ni, Yichao Yan, Peng Zhou, Shuo Cheng, and Jianguo Hu. Pose transferrable person re-identification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4099–4108, 2018.
- [120] Jun Liu, Chao Liang, Mang Ye, Zheng Wang, Yang Yang, Zhen Han, and Kaimin Sun. Person re-identification via attribute confidence and saliency. In *Pacific Rim Conference on Multimedia*, pages 591–600. Springer, 2015.
- [121] Si Liu, Zheng Song, Guangcan Liu, Changsheng Xu, Hanqing Lu, and Shuicheng Yan. Street-to-shop: Cross-scenario clothing retrieval via parts alignment and auxiliary set. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3330–3337. IEEE, 2012.
- [122] Si Liu, Jiashi Feng, Csaba Domokos, Hui Xu, Junshi Huang, Zhenzhen Hu, and Shuicheng Yan. Fashion parsing with weak color-category labels. *IEEE Transactions on Multimedia*, 16(1):253–265, 2014.
- [123] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.
- [124] Chen Change Loy, Tao Xiang, and Shaogang Gong. Multi-camera activity correlation analysis. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1988–1995. IEEE, 2009.
- [125] Chen Change Loy, Tao Xiang, and Shaogang Gong. Time-delayed correlation analysis for multi-camera activity understanding. *International Journal of Computer Vision*, 90(1):106–129, 2010.
- [126] Chen Change Loy, Chunxiao Liu, and Shaogang Gong. Person re-identification by manifold ranking. In *2013 IEEE International Conference on Image Processing*, pages 3567–3571. IEEE, 2013.

- [127] Saikat Maitra and Jun Yan. Principle component analysis and partial least squares: Two dimension reduction techniques for regression. *Applying Multivariate Statistical Models*, 79:79–90, 2008.
- [128] Melvin Earl Maron. Automatic indexing: an experimental inquiry. *Journal of the ACM (JACM)*, 8(3):404–417, 1961.
- [129] Tetsu Matsukawa and Einoshin Suzuki. Person re-identification using cnn features learned from combination of attributes. In *2016 23rd International Conference on Pattern Recognition (ICPR)*, pages 2428–2433. IEEE, 2016.
- [130] Tetsu Matsukawa, Takahiro Okabe, Einoshin Suzuki, and Yoichi Sato. Hierarchical gaussian descriptor for person re-identification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1363–1372, 2016.
- [131] Warren S McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133, 1943.
- [132] Niall McLaughlin, Jesus Martinez del Rincon, and Paul C Miller. Person reidentification using deep convnets with multitask learning. *IEEE Transactions on Circuits and Systems for Video Technology*, 27(3):525–539, 2016.
- [133] Laurence Meylan and Sabine Süsstrunk. Color image enhancement using a retinex-based adaptive filter. In *Conference on Colour in Graphics, Imaging, and Vision*, volume 2004, number 1, pages 359–363. Society for Imaging Science and Technology, 2004.
- [134] Marvin Minsky. Steps toward artificial intelligence. *Proceedings of the IRE*, 49(1):8–30, 1961.
- [135] Matteo Munaro, Alberto Basso, Andrea Fossati, Luc Van Gool, and Emanuele Menegatti. 3d reconstruction of freely moving persons for re-identification with a depth sensor. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4512–4519. IEEE, 2014.
- [136] Matteo Munaro, Stefano Ghidoni, Deniz Tartaro Dizmen, and Emanuele Menegatti. A feature-based approach to people re-identification using skeleton keypoints. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5644–5651. IEEE, 2014.

- [137] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 807–814, 2010.
- [138] Timo Ojala, Matti Pietikäinen, and David Harwood. A comparative study of texture measures with classification based on featured distributions. *Pattern recognition*, 29(1):51–59, 1996.
- [139] OpenNI, 2010. URL <http://www.openni.org>.
- [140] OpenNI, 2010. URL <http://www.openni.org/files/nite/>.
- [141] Foram S Panchal and Mahesh Panchal. Review on methods of selecting number of hidden nodes in artificial neural network. *International Journal of Computer Science and Mobile Computing*, 3(11):455–464, 2014.
- [142] Constantine Papageorgiou and Tomaso Poggio. A trainable system for object detection. *International journal of computer vision*, 38(1):15–33, 2000.
- [143] Dabal Pedamonti. Comparison of non-linear activation functions for deep neural networks on mnist classification task. *arXiv preprint arXiv:1804.02763*, 2018.
- [144] Peixi Peng, Tao Xiang, Yaowei Wang, Massimiliano Pontil, Shaogang Gong, Tiejun Huang, and Yonghong Tian. Unsupervised cross-dataset transfer learning for person re-identification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1306–1315, 2016.
- [145] Ana Belén Petro, Catalina Sbert, and Jean-Michel Morel. Multiscale retinex. *Image Processing On Line*, pages 71–88, 2014.
- [146] Víctor Ponce-López, Tilo Burghardt, Sion Hannunna, Dima Damen, Alessandro Masullo, and Majid Mirmehdi. Semantically selective augmentation for deep compact person re-identification. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 0–0, 2018.
- [147] Lorien Y Pratt. Discriminability-based transfer between neural networks. In *Advances in neural information processing systems*, pages 204–211, 1993.
- [148] Bryan James Prosser, Wei-Shi Zheng, Shaogang Gong, Tao Xiang, and Q Mary. Person re-identification by support vector ranking. In *BMVC*, volume 2, number 5, page 6, 2010.

- [149] Xuelin Qian, Yanwei Fu, Tao Xiang, Wenxuan Wang, Jie Qiu, Yang Wu, Yu-Gang Jiang, and Xiangyang Xue. Pose-normalized image generation for person re-identification. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 650–667, 2018.
- [150] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.
- [151] Sashank J Reddi, Satyen Kale, and Sanjiv Kumar. On the convergence of adam and beyond. *arXiv preprint arXiv:1904.09237*, 2019.
- [152] Ergys Ristani, Francesco Solera, Roger Zou, Rita Cucchiara, and Carlo Tomasi. Performance measures and a data set for multi-target, multi-camera tracking. In *European Conference on Computer Vision*, pages 17–35. Springer, 2016.
- [153] Frank Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386, 1958.
- [154] Frank Rosenblatt. Principles of neurodynamics. perceptrons and the theory of brain mechanisms. Technical report, Cornell Aeronautical Lab Inc Buffalo NY, 1961.
- [155] Roman Rosipal and Nicole Krämer. Overview and recent advances in partial least squares. In *International Statistical and Optimization Perspectives Workshop "Subspace, Latent Structure and Feature Selection"*, pages 34–51. Springer, 2005.
- [156] Peter M Roth, Martin Hirzer, Martin Köstinger, Csaba Beleznai, and Horst Bischof. Mahalanobis distance learning for person re-identification. In *Person re-identification*, pages 247–267. Springer, 2014.
- [157] Peter Sadowski. Notes on backpropagation. *homepage: <https://www.ics.uci.edu/pjsadows/notes.pdf> (online)*, 2016.
- [158] Warren S Sarle. Stopped training and other remedies for overfitting. *Computing science and statistics*, pages 352–360, 1996.
- [159] Cordelia Schmid. Constructing models for content-based image retrieval. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, volume 2, pages II–II. IEEE, 2001.

- [160] Jürgen Schmidhuber. Deep learning in neural networks: An overview. *Neural networks*, 61:85–117, 2015.
- [161] Bernhard Schölkopf, Alexander J Smola, Francis Bach, et al. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press, 2002.
- [162] Arne Schumann and Rainer Stiefelhagen. Person re-identification by deep learning attribute-complementary information. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 20–28, 2017.
- [163] Shiven Sharma, Colin Bellinger, Bartosz Krawczyk, Osmar Zaiane, and Nathalie Japkowicz. Synthetic oversampling with the majority class: A new perspective on handling extreme imbalance. In *2018 IEEE International Conference on Data Mining (ICDM)*, pages 447–456. IEEE, 2018.
- [164] Zhiyuan Shi, Timothy M Hospedales, and Tao Xiang. Transferring a semantic representation for person re-identification and search. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4184–4193, 2015.
- [165] Hoo-Chang Shin, Holger R Roth, Mingchen Gao, Le Lu, Ziyue Xu, Isabella Noguez, Jianhua Yao, Daniel Mollura, and Ronald M Summers. Deep convolutional neural networks for computer-aided detection: Cnn architectures, dataset characteristics and transfer learning. *IEEE transactions on medical imaging*, 35(5):1285–1298, 2016.
- [166] Jamie Shotton, Andrew Fitzgibbon, Mat Cook, Toby Sharp, Mark Finocchio, Richard Moore, Alex Kipman, and Andrew Blake. Real-time human pose recognition in parts from single depth images. In *CVPR 2011*, pages 1297–1304. Ieee, 2011.
- [167] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.
- [168] Josef Sivic and Andrew Zisserman. Efficient visual search of videos cast as text retrieval. *IEEE transactions on pattern analysis and machine intelligence*, 31(4):591–606, 2009.
- [169] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks

- from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.
- [170] Chi Su, Fan Yang, Shiliang Zhang, Qi Tian, Larry S Davis, and Wen Gao. Multi-task learning with low rank attribute embedding for person re-identification. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3739–3747, 2015.
- [171] Chi Su, Shiliang Zhang, Junliang Xing, Wen Gao, and Qi Tian. Deep attributes driven multi-camera person re-identification. In *European conference on computer vision*, pages 475–491. Springer, 2016.
- [172] Chi Su, Jianing Li, Shiliang Zhang, Junliang Xing, Wen Gao, and Qi Tian. Pose-driven deep convolutional model for person re-identification. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3960–3969, 2017.
- [173] Chi Su, Shiliang Zhang, Fan Yang, Guangxiao Zhang, Qi Tian, Wen Gao, and Larry S Davis. Attributes driven tracklet-to-tracklet person re-identification using latent prototypes space mapping. *Pattern Recognition*, 66:4–15, 2017.
- [174] Chi Su, Shiliang Zhang, Junliang Xing, Wen Gao, and Qi Tian. Multi-type attributes driven multi-camera person re-identification. *Pattern Recognition*, 75:77–89, 2018.
- [175] Liang Sun, Shuiwang Ji, Shipeng Yu, and Jieping Ye. On the equivalence between canonical correlation analysis and orthonormalized partial least squares. In *Twenty-First International Joint Conference on Artificial Intelligence*, 2009.
- [176] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.
- [177] Tijmen Tieleman and Geoffrey Hinton. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural networks for machine learning*, 4(2):26–31, 2012.
- [178] A.N. Tikhonov. On the stability of inverse problems. *Doklady Akademii nauk SSSR*, 39(5):195–198, 1943.
- [179] Lisa Torrey and Jude Shavlik. Transfer learning. In *Handbook of research on machine learning applications and trends: algorithms, methods, and techniques*, pages 242–264. IGI Global, 2010.

- [180] Andrew Turpin and Falk Scholer. User performance versus precision measures for simple search tasks. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 11–18. ACM, 2006.
- [181] Joost Van De Weijer, Cordelia Schmid, Jakob Verbeek, and Diane Larlus. Learning color names for real-world applications. *IEEE Transactions on Image Processing*, 18(7):1512–1523, 2009.
- [182] Jan N van Rijn and Frank Hutter. Hyperparameter importance across datasets. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2367–2376. ACM, 2018.
- [183] Chunyu Wang, Yizhou Wang, Zhouchen Lin, Alan L Yuille, and Wen Gao. Robust estimation of 3d human poses from a single image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2361–2368, 2014.
- [184] Jin Wang, Zheng Wang, Changxin Gao, Nong Sang, and Rui Huang. Deeplist: Learning deep features with adaptive listwise constraint for person reidentification. *IEEE Transactions on Circuits and Systems for Video Technology*, 27(3): 513–524, 2017.
- [185] Jin Wang, Zheng Wang, Chao Liang, Changxin Gao, and Nong Sang. Equidistance constrained metric learning for person re-identification. *Pattern Recognition*, 74:38–51, 2018.
- [186] Jingya Wang, Xiatian Zhu, Shaogang Gong, and Wei Li. Transferable joint attribute-identity deep learning for unsupervised person re-identification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2275–2284, 2018.
- [187] Taiqing Wang, Shaogang Gong, Xiatian Zhu, and Shengjin Wang. Person re-identification by video ranking. In *European Conference on Computer Vision*, pages 688–703. Springer, 2014.
- [188] Xiaogang Wang and Rui Zhao. Person re-identification: System design and evaluation overview. In *Person Re-Identification*, pages 351–370. Springer, 2014.
- [189] Gregory Watson and Abhir Bhalerao. Person re-identification using partial least squares appearance modelling. In *International Conference on Image Analysis and Processing*, pages 25–36. Springer, 2017. doi: 10.1007/978-3-319-68548-9_3.

- [190] Gregory Watson and Abhir Bhalerao. Person reidentification using deep foreground appearance modeling. *Journal of Electronic Imaging*, 27(5):051215, 2018. doi: 10.1117/1.JEI.27.5.051215.
- [191] Gregory Watson and Abhir Bhalerao. Person re-identification combining deep features and attribute detection. *Multimedia Tools and Applications*, December 2019. doi: 10.1007/s11042-019-08499-9.
- [192] Longhui Wei, Shiliang Zhang, Wen Gao, and Qi Tian. Person transfer gan to bridge domain gap for person re-identification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 79–88, 2018.
- [193] Shih-En Wei, Varun Ramakrishna, Takeo Kanade, and Yaser Sheikh. Convolutional pose machines. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4724–4732, 2016.
- [194] Kilian Q Weinberger and Lawrence K Saul. Fast solvers and efficient implementations for distance metric learning. In *Proceedings of the 25th international conference on Machine learning*, pages 1160–1167. ACM, 2008.
- [195] Kilian Q Weinberger, John Blitzer, and Lawrence K Saul. Distance metric learning for large margin nearest neighbor classification. In *Advances in neural information processing systems*, pages 1473–1480, 2006.
- [196] Svante Wold, Arnold Ruhe, Herman Wold, and WJ Dunn, III. The collinearity problem in linear regression. the partial least squares (pls) approach to generalized inverses. *SIAM Journal on Scientific and Statistical Computing*, 5(3): 735–743, 1984.
- [197] Svante Wold, Kim Esbensen, and Paul Geladi. Principal component analysis. *Chemometrics and intelligent laboratory systems*, 2(1-3):37–52, 1987.
- [198] Svante Wold, Michael Sjöström, and Lennart Eriksson. Pls-regression: a basic tool of chemometrics. *Chemometrics and intelligent laboratory systems*, 58(2): 109–130, 2001.
- [199] Ancong Wu, Wei-Shi Zheng, and Jian-Huang Lai. Robust depth-based person re-identification. *IEEE Transactions on Image Processing*, 26(6):2588–2603, 2017.
- [200] Lin Wu, Chunhua Shen, and Anton van den Hengel. Personnet: Person re-identification with deep convolutional neural networks. *arXiv preprint arXiv:1601.07255*, 2016.

- [201] Lin Wu, Chunhua Shen, and Anton Van Den Hengel. Deep linear discriminant analysis on fisher networks: A hybrid architecture for person re-identification. *Pattern Recognition*, 65:238–250, 2017.
- [202] Qiong Wu, Pingyang Dai, Peixian Chen, and Yuyu Huang. Deep adversarial data augmentation with attribute guided for person re-identification. *Signal, Image and Video Processing*, pages 1–8, 2019.
- [203] Shangxuan Wu, Ying-Cong Chen, Xiang Li, An-Cong Wu, Jin-Jie You, and Wei-Shi Zheng. An enhanced deep feature representation for person re-identification. In *2016 IEEE winter conference on applications of computer vision (WACV)*, pages 1–8. IEEE, 2016.
- [204] Yuxin Wu et al. Tensorpack. <https://github.com/tensorpack/>, 2016.
- [205] Qiqi Xiao, Kelei Cao, Haonan Chen, Fangyue Peng, and Chi Zhang. Cross domain knowledge transfer for person re-identification. *arXiv preprint arXiv:1611.06026*, 2016.
- [206] Tong Xiao, Hongsheng Li, Wanli Ouyang, and Xiaogang Wang. Learning deep feature representations with domain guided dropout for person re-identification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1249–1258, 2016.
- [207] Bing Xu, Naiyan Wang, Tianqi Chen, and Mu Li. Empirical evaluation of rectified activations in convolutional network. *arXiv preprint arXiv:1505.00853*, 2015.
- [208] Yang Yang, Jimei Yang, Junjie Yan, Shengcai Liao, Dong Yi, and Stan Z Li. Salient color names for person re-identification. In *European conference on computer vision*, pages 536–551. Springer, 2014.
- [209] Xin Ye, Wen-yuan Zhou, and Lu-an Dong. Body part-based person re-identification integrating semantic attributes. *Neural Processing Letters*, 49(3): 1111–1124, 2019.
- [210] Özgür Yeniay and Atilla Göktaş. A comparison of partial least squares regression with other prediction methods. *Hacettepe Journal of Mathematics and Statistics*, 31:99–111, 2002.
- [211] Zhou Yin, Wei-Shi Zheng, Ancong Wu, Hong-Xing Yu, Hai Wan, Xiaowei Guo, Feiyue Huang, and Jianhuang Lai. Adversarial attribute-image person re-identification. *arXiv preprint arXiv:1712.01493*, 2017.

- [212] Li Zhang, Tao Xiang, and Shaogang Gong. Learning a discriminative null space for person re-identification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1239–1248, 2016.
- [213] Ying Zhang, Baohua Li, Huchuan Lu, Atshushi Irie, and Xiang Ruan. Sample-specific svm learning for person re-identification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1278–1287, 2016.
- [214] Cairong Zhao, Kang Chen, Zhihua Wei, Yipeng Chen, Duoqian Miao, and Wei Wang. Multilevel triplet deep learning model for person re-identification. *Pattern Recognition Letters*, 117:161–168, 2019.
- [215] Liming Zhao, Xi Li, Yueting Zhuang, and Jingdong Wang. Deeply-learned part-aligned representations for person re-identification. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3219–3228, 2017.
- [216] Rui Zhao, Wanli Ouyang, and Xiaogang Wang. Unsupervised salience learning for person re-identification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3586–3593, 2013.
- [217] Rui Zhao, Wanli Ouyang, and Xiaogang Wang. Learning mid-level filters for person re-identification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 144–151, 2014.
- [218] Yiru Zhao, Xu Shen, Zhongming Jin, Hongtao Lu, and Xian-sheng Hua. Attribute-driven feature disentangling and temporal aggregation for video person re-identification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4913–4922, 2019.
- [219] Liang Zheng, Liyue Shen, Lu Tian, Shengjin Wang, Jingdong Wang, and Qi Tian. Scalable person re-identification: A benchmark. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1116–1124, 2015.
- [220] Liang Zheng, Zhi Bie, Yifan Sun, Jingdong Wang, Chi Su, Shengjin Wang, and Qi Tian. Mars: A video benchmark for large-scale person re-identification. In *European Conference on Computer Vision*, pages 868–884. Springer, 2016.
- [221] Liang Zheng, Yi Yang, and Alexander G Hauptmann. Person re-identification: Past, present and future. *arXiv preprint arXiv:1610.02984*, 2016.
- [222] Liang Zheng, Yujia Huang, Huchuan Lu, and Yi Yang. Pose invariant embedding for deep person re-identification. *arXiv preprint arXiv:1701.07732*, 2017.

- [223] Liang Zheng, Hengheng Zhang, Shaoyan Sun, Manmohan Chandraker, Yi Yang, and Qi Tian. Person re-identification in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1367–1376, 2017.
- [224] Meng Zheng, Srikrishna Karanam, and Richard J Radke. Rpifield: A new dataset for temporally evaluating person re-identification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 1893–1895, 2018.
- [225] Wei-Shi Zheng, Shaogang Gong, and Tao Xiang. Associating groups of people. In *BMVC*, 2009.
- [226] Wei-Shi Zheng, Shaogang Gong, and Tao Xiang. Person re-identification by probabilistic relative distance comparison. In *CVPR 2011*, pages 649–656. IEEE, 2011.
- [227] Wei-Shi Zheng, Shaogang Gong, and Tao Xiang. Reidentification by relative distance comparison. *IEEE transactions on pattern analysis and machine intelligence*, 35(3):653–668, 2013.
- [228] Zhedong Zheng, Liang Zheng, and Yi Yang. Unlabeled samples generated by gan improve the person re-identification baseline in vitro. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3754–3762, 2017.
- [229] Zhedong Zheng, Liang Zheng, and Yi Yang. A discriminatively learned cnn embedding for person reidentification. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 14(1):13, 2018.
- [230] Weilin Zhong, Linfeng Jiang, Tao Zhang, Jinsheng Ji, and Huilin Xiong. Combining multilevel feature extraction and multi-loss learning for person re-identification. *Neurocomputing*, 334:68–78, 2019.
- [231] Zhun Zhong, Liang Zheng, Zhedong Zheng, Shaozi Li, and Yi Yang. Camera style adaptation for person re-identification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5157–5166, 2018.