ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL OF SCIENCE ENGINEERING AND TECHNOLOGY

DEEP CONVOLUTIONAL NEURAL NETWORK BASED REPRESENTATIONS FOR PERSON RE-IDENTIFICATION

M.Sc. THESIS

Alper ULU

Department of Computer Engineering

Computer Engineering Programme

JUNE 2016

ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL OF SCIENCE ENGINEERING AND TECHNOLOGY

DEEP CONVOLUTIONAL NEURAL NETWORK BASED REPRESENTATIONS FOR PERSON RE-IDENTIFICATION

M.Sc. THESIS

Alper ULU
(504131503)

Department of Computer Engineering

Computer Engineering Programme

Thesis Advisor: Assoc. Prof. Dr. Hazım Kemal EKENEL

JUNE 2016

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

KİŞİYİ YENİDEN TANIMA İÇİN DERİN EVRİŞİMSEL SİNİR AĞI TABANLI MODELLER

YÜKSEK LİSANS TEZİ

Alper ULU
(504131503)

Bilgisayar Mühendisliği Anabilim Dalı

Bilgisayar Mühendisliği Programı

Tez Danışmanı: Assoc. Prof. Dr. Hazım Kemal EKENEL

HAZİRAN 2016

Alper ULU, a M.Sc. student of ITU Graduate School of Science Engineering and Technology 504131503 successfully defended the thesis entitled "DEEP CONVOLUTIONAL NEURAL NETWORK BASED REPRESENTATIONS FOR PERSON RE-IDENTIFICATION", which he/she prepared after fulfilling the requirements specified in the associated legislations, before the jury whose signatures are below.

Thesis Advisor :	**Assoc. Prof. Dr. Hazım Kemal EKENEL** Istanbul Technical University	.....
Jury Members :	**Prof. Dr. Muhittin Gökmen** MEF University	.....
	Assoc. Prof. Dr. Gözde Ünal Istanbul Technical University	.....
		.....

Date of Submission : **2 May 2016**
Date of Defense : **6 June 2016**

To my family,

FOREWORD

I would like to thank to my supervisor, Assoc. Prof. Dr. Hazım Kemal EKENEL for his guidance throughout my research process and during the preperation of this thesis.

Special thanks to my family, for their patience and continuous support during my master study and during the preparation of this work.

And finally, I would like to thank all SiMiT Lab members for valuable discussions and for sharing their knowledge with me.

June 2016

Alper ULU
Computer Engineer

TABLE OF CONTENTS

Page

FOREWORD ..... **ix**
TABLE OF CONTENTS ..... **xi**
ABBREVIATIONS ..... **xiii**
LIST OF TABLES ..... **xv**
LIST OF FIGURES ..... **xvii**
SUMMARY ..... **xix**
ÖZET ..... **xxi**
1. INTRODUCTION **1**
1.1 Purpose of Thesis 4
1.2 Literature Review 4
1.3 Hypothesis 7
2. AN OVERVIEW OF CONVOLUTIONAL NEURAL NETWORKS **9**
2.1 Components 9
2.1.1 Convolutional layer 10
2.1.2 Activation functions 10
2.1.3 Pooling 11
2.1.4 Dropout-DropConnect 12
2.1.5 Fully connected layer 13
2.1.6 Output layer 13
2.2 Training a Convolutional Neural Network 13
2.2.1 Backpropagation algorithm 13
2.2.1.1 Feedforward 14
2.2.1.2 Backpropagation 15
2.3 Off-the-Shelf Deep Neural Network Architectures 16
2.3.1 AlexNet 16
2.3.2 VGG-16 18
2.3.3 GoogLeNet 19
3. TRANSFER LEARNING ON CONVOLUTIONAL NEURAL NETWORKS **21**
3.1 Transfer Learning in a Nutshell 21
3.2 Fine-Tuning the Convolutional Neural Networks 22
3.3 Fine-tuning Results 23
3.3.1 Fine-tuning on AlexNet 23
3.3.2 Fine-tuning on VGG-16 25
3.3.3 Fine-tuning on GoogleNet 27
4. FEATURE EXTRACTION AND METRIC LEARNING **29**
4.1 Feature Extraction from Fine-Tuned Networks 29

4.2 Metric Learning Methods 31
4.2.1 Large margin dimensionality reduction 31
4.2.2 Joint metric learning 32
4.2.3 Diagonal metric learning 33
5. CLASSIFICATION 35
5.1 K-Nearest Neighbors Algorithm 35
5.2 Distance Metrics 36
6. EXPERIMENTAL RESULTS 37
6.1 Datasets 37
6.2 Pipeline of the Proposed Method 39
6.2.1 Preprocessing 39
6.2.2 Training 42
6.2.3 Test 42
6.3 Results 43
6.3.1 Metric learning results 43
6.3.2 Distance metric results 45
6.3.2.1 Results on VIPeR dataset 45
6.3.2.2 Results on PRID 2011 dataset 47
6.3.2.3 Results on iLIDS-VID dataset 47
7. CONCLUSIONS AND FUTURE RECOMMENDATIONS 49
REFERENCES 51
APPENDICES 55
APPENDIX A.1 57
APPENDIX A.2 63
CURRICULUM VITAE 65

ABBREVIATIONS

BP	**:** Backpropagation
BBf	**:** Best Bin First
CNN	**:** Convolutional Neural Network
CMC	**:** Cummulative Matching Curve
DML	**:** Deep Metric Learning
HOG	**:** Histogram Of Gradients
HSV	**:** Hue Saturation Value
ITML	**:** Information Theoretic Metric Learning algorithm
KNN	**:** k-Nearest Neighbour
KD-Tree	**:** K Dimensional Tree
KISSME	**:** Keep It Simple and Straightforward Metric
LMNN	**:** Large Margin Nearest Neighbor
LBP	**:** Local Binary Patterns
LADF	**:** Locally-Adaptive Decision Functions
PCH	**:** Probabilistic Color Histogram
RGB	**:** Red Green Blue
SIFT	**:** Scale Invariant Feature Transform
SURF	**:** Speeded Up Robust Features
SGD	**:** Stochastic Gradient Descent
SVM	**:** Support Vector Machines

LIST OF TABLES

Page

Table 3.1 : Fine-tunig accuracy results for AlexNet [19]. 25
Table 3.2 : Fine-tuning accuracy results for VGG-16 [20]. 27
Table 3.3 : Fine-tune accuracy results for GoogLeNet [21]. 28
Table 6.1 : Fused results with different Rank (r) values on VIPeR [9]. 45
Table 6.2 : Comparison with the other methods on VIPeR [9]. 46
Table 6.3 : Fused results with different Rank (r) values on PRID 2011 [23]. 47
Table 6.4 : Fused results with different Rank (r) values on iLIDS-VID [24]. 48

LIST OF FIGURES

Page

Figure 1.1 : Multidimensional taxonomy for people re-identification algorithms [2]. 2
Figure 1.2 : Pipeline of person re-identification problem [4]. 4
Figure 2.1 : The complete implementation of the LeNet-5 [26]. 9
Figure 2.2 : Convolutional layer. 10
Figure 2.3 : The most used activation functions in neural networks [28]. 11
Figure 2.4 : Max pooling with a 2x2 filter and stride = 2 [29]. 11
Figure 2.5 : Left: Dropout, right: DropConnect [27]. 12
Figure 2.6 : An example neural network with one hidden layer [32]. 14
Figure 2.7 : AlexNet architecture [19]. 17
Figure 2.8 : VGG-16 network architecture [20]. 18
Figure 2.9 : GoogLeNet network architecture [21]. 20
Figure 3.1 : Traditional learning and transfer learning [35]. 22
Figure 3.2 : Fine-tuning procedure for each body part. 24
Figure 3.3 : Fine-tuning procedure for full body. 24
Figure 3.4 : New learning rates for AlexNet [19]. 25
Figure 3.5 : AlexNet [19] fine-tuning results for CUHK03 [4] dataset. 26
Figure 3.6 : VGG-16 [20] fine-tuning results for CUHK03 [4] dataset. 27
Figure 3.7 : CUHK03 [4] fine-tuning results for VGG-16 [20]. 28
Figure 4.1 : Person representation. 30
Figure 4.2 : Person representation. 30
Figure 4.3 : Positive and negative person pairs are seperated with a margin by metric learning. 32
Figure 5.1 : k-nn algorithm for k=3. 35
Figure 6.1 : (a) Example image pairs from the PRID 2011 [23], (b) the iLIDS-VID (b) [24], and (c) the VIPeR [9]. 39
Figure 6.2 : Pipeline of our proposed method. 40
Figure 6.3 : Prerocessing steps: **a)**original image **b)**histogram equalized image **c)**ellipse mask applied image **d)**division into three part. 42
Figure 6.4 : Joint metric results on VIPeR [9]. 44
Figure 6.5 : Diagonal metric results on VIPeR [9]. 44
Figure 6.6 : The CMC curve on the VIPeR [9] dataset, the training sets are CUHK03 [4] and Market-1501 [22]. 46
Figure 6.7 : The CMC curve on the PRID [23] dataset. 48
Figure 6.8 : The CMC curve on the iLIDS-VID dataset. 48
Figure A.1 : AlexNet [19] fine-tuning results for Market-1501 [22] dataset. 57

Figure A.2 : AlexNet [19] fine-tuning results for the union of Market-1501 [22] and CUHK03 [4] datasets. 58
Figure A.3 : VGG-16 [20] fine-tuning results for Market-1501 [22] dataset. 59
Figure A.4 : VGG-16 [20] fine-tuning results for the union of Market-1501 [22] and CUHK03 [4] datasets. 60
Figure A.5 : GoogLeNet [21] fine-tuning results for Market-1501 [22] dataset. .. 61
Figure A.6 : GoogLeNet [21] fine-tuning results for the union of Market-1501 [22] and CUHK03 [4] datasets. 62
Figure B.1 : Joint metric results on VIPeR [9] for head part. 63
Figure B.2 : Joint metric results on VIPeR [9] for body part. 63
Figure B.3 : Joint metric results on VIPeR [9] for leg part. 63
Figure B.4 : Diagonal metric results on VIPeR [9] for head part. 64
Figure B.5 : Diagonal metric results on VIPeR [9] for body part. 64
Figure B.6 : Diagonal metric results on VIPeR [9] for leg part. 64

DEEP CONVOLUTIONAL NEURAL NETWORK BASED REPRESENTATIONS FOR PERSON RE-IDENTIFICATION

SUMMARY

Video surveillance systems have great importance to ensure public safety. Today, these kind of systems not only capture and distribute video but also have so many smart applications. Their main task is to detect abnormal events and prevent crimes before they happen. Thanks to having these features, they can be very beneficial to help security attendants. Because, monitoring a large camera network is a very labor-intensive task. These smart applications can generate real time alarms to call security staff's attention. Person re-identification mechanism is one of these applications. It has wide usage area and very important to find suspected persons.

Person re-identification problem can be defined as matching pedestrian images which are obtained from different video cameras. This is a very challenging task and it may contain many parameters. Differences in lighting conditions, background changes, occlusions, camera angles and pose variations make the problem even harder. Also, the problem scenario may contain many variations: we can have one or more images for each person, we can have very different camera combinations or we can have different datasets etc. So, person re-identification is still an open problem and there is no prominent solution to solve all different scenarios.

In this thesis, we have proposed a deep learning and metric learning based method for cross dataset person re-identification problem. Firstly, cross dataset means that, we used different datasets at training and testing stages. Until now, most of the proposed methods have concentrated on single dataset setting. Secondly, deep learning based approaches have achieved great results in many computer vision problems such as digit recognition, scene understanding and face verification. Person re-identification is one of them and quite good results have been published. We used several convolutional neural network architectures in our experiments and we took advantage of their good feature extraction power.

The success rate of deep learning based solutions is highly related to the size of training data that you have. Because of this reason, we used the largest datasets, which are prepared for person re-identification problem. However, these datasets are not enough to train a whole network from scratch. It is highly recommended that, your training data size should be at the level of millions to train a convolutional neural network. At this point, transfer learning procedure is a good option that should be considered. In this thesis, our main contribution is to show that some well-performing domain specific convolutional neural networks can be used in also person re-identification problem. To do this, we selected some neural networks which are good at image recognition problem and fine-tuned them with the largest person re-identification datasets. While we were doing this, we kept early layers weights as much as possible and we mostly

changed the last layer weights according to our problem. Their accuracy-loss results showed that these networks have pretty good learning capabilities for our problem. In this thesis, AlexNet, VGG-16 and GoogLeNet convolutional neural networks have been used for feature extraction. We fine-tuned these neural networks for each human body part separately.

After fine-tuning operation, the neural networks have become ready to extract good features from person images. At this point, we followed two different ways. First one is metric learning methods and the other one is direct similarity calculation on test set. For the metric learning, we have extracted our training set features from convolutional neural networks which are just fine-tuned. While we were doing this, we used different layers of neural networks. Next, we prepared positive and negative image pairs to be used in metric learning step. Here, positive and negative pairs mean that whether two images belongs to same person or not. Because, pairwise metric learning methods use this information and produce a projection matrix. This matrix moves the features from their current space to more discriminative feature space. In that domain, same person's images are relatively closer to the each other. At the test stage, we used this matrix with test features and project them to their new domain.

Second evaluation method is direct similarity comparison of test features. After features extracted from head, body and leg parts, we fused them to obtain final person representation and calculated similarity between probe and gallery images. Here, we followed a common test procedure to evaluate our success rate. For similarity measurement, we used cosine distance metric. In that distance, low values mean that these features are closer to the each other and also they are more similar. We have achieved %32 matching rate at Rank-1 value. This result is one of the best result for cross dataset person re-identification problem. We also compared our results with other approaches which are published for this problem. All results have been drawn as a Cummulative Matching Curves.

KİŞİYİ YENİDEN TANIMA İÇİN DERİN EVRİŞİMSEL SİNİR AĞI TABANLI MODELLER

ÖZET

Günümüzde video güvenlik sistemleri kamu güvenliğini sağlama konusunda büyük bir önem taşımaktadır. Hemen her yerde görebileceğimiz bu sistemlerin barındırdığı kamera sayısı oldukça yüksek sayıda olabilmektedir. Genellikle tek bir noktadan takip edilen bu sistemler, görevli kişiler tarafından gün boyunca izlenmekte ve kaydedilmektedir. Ancak gün boyu farklı kameradan gelen bu görüntülerin takip edilmesi oldukça yoğun dikkat isteyen yorucu bir iştir. Bu sebeplerle, günümüzde bu sistemler sadece görüntü kaydetmek ve dağıtmakla kalmayıp aynı zamanda çeşitli akıllı uygulamalar da barındırır hale gelmiştir. Bu uygulamaların amacı şüpheli olay ve hareketleri tespit etmek ve olabildiğince erken uyarıyı vererek, görevli kişilere yardımcı olmaktır. Yüz tanıma, kişi takibi, şüpheli paket tespiti, tehlikeli aktivitelerin belirlenmesi konusunda çalışan uygulamalar bunların başlıcalarıdır. Kişinin yeniden tanınması problemi de bu alanda kullanılan uygulamalardan bir tanesidir ve günümüzde oldukça yüksek bir öneme sahiptir.

Kişiyi yeniden tanıma problemi, farklı kameralardan sağlanan görüntülerin, aynı bireye ait olup olmadığının belirlenmesi olarak tanımlanabilir. Bu problem, bilgisayarla görü alanındaki zorlu araştırma konularından birisidir. Işıklandırma koşulları, farklı kamera açıları, poz değişimleri, arka plan değişimleri ve kameraların düşük çözünürlükte olması gibi çeşitli dış faktörler, problemi zorlaştırmaktadırlar. Ayrıca problemi oluşturabilecek çok farklı senaryo bulunmaktadır. Örneğin; kişilere ait birer resmin veya video kaydının bulunması, kamera açılarının kesişmesi veya birbirinden bağımsız ortamlardan elde edilen görüntüler bunlardan bazılarıdır. Kişiyi yeniden tanıma problemi hala çözümlere açık bir problemdir. Şu ana kadar kesin bir çözümünün elde edilememesi nedeni ile bu problem üzerindeki çalışmalar artarak devam etmektedir.

Özellikle etiketli veri miktarındaki artış ve GPU tabanlı teknolojilerdeki gelişmelerle beraber derin öğrenme tabanlı çözümler, bilgisayarla görü ve makine öğrenmesi problemlerinde büyük başarılar elde etmektedir. Belirli bir süre eğitim aşamasına ihtiyaç duyan derin öğrenme tabanlı sistemlerde başarım, eğitim kümesinin büyüklüğü ve kullanılan evrişimsel sinir ağının derinliği ile doğru orantılıdır. Kişiyi yeniden tanıma problemi için kullanılan veri kümelerinin büyümesi, derin öğrenme tabanlı çözümlerin klasik öznitelik çıkarımı tabanlı sistemleri geride bırakmasını sağlamıştır. Eğitim kümesinin yeterli düzeyde olmadığı problemlerde transfer öğrenme yaklaşımı, büyük kümelerle eğitimi sağlanmış sinir ağlarından elde edilen bilgi birikiminin ihtiyaç duyulan problemlere aktarılmasına olanak tanımaktadır.

Bu çalışmada, kişiyi yeniden tanıma problemi için derin öğrenme ve metrik öğrenme tabanlı bir çözüm önerilmiştir. Bunun için şu anda bilinen en büyük kişiyi yeniden tanıma eğitim kümeleri kullanılmıştır. Bu eğitim kümeleri oldukça büyük olmasına

rağmen, bir evrişimsel sinir ağını sıfır ağırlıkları ile en başından itibaren eğitecek düzeyde değildir. Bu işlem için kullanılacak eğitim setlerinin büyüklüğünün en az milyon seviyesinde olması tavsiye edilmektedir. Bu nedenle bu çalışmada transfer eğitim metodu ile farklı bir alanda oldukça başarılı sonuçlar veren sinir ağlarından faydalanıldı. Görüntü tanıma probleminde oldukça başarılı sonuçlar elde eden bu ağlar kişiyi yeniden tanıma problemine özgü eğitim setleri ile ince ayar işlemine tabi tutuldu. Böylece kendi problemlerinde başarılı evrişimsel sinir ağları, çeşitli ayarlamalar ile kişiyi yeniden tanıma problemine uygun hale getirildi.

Evrişimsel sinir ağları birbirinden farklı görevlere sahip olan sıralı bir dizi katmandan oluşmaktadır. Evrişim katmanı da bunlardan bir tanesidir. Bu katman, görüntü üzerinde sahip olduğu büyüklük kadar iki veya üç boyutlu evrişim işlemini gerçekleştirerek, sinir ağı boyunca probleme özgü ayırt edici filtrelerin oluşmasını sağlamaktadır. Bir evrişimsel sinir ağındaki ilk evrişim katmanları, görüntü üzerinden daha genel özniteliklerin çıkarılmasını sağlamaktadır. Örnek olarak bu katmanlar; kenar bulma, köşe bulma, bölge bulma filtreleri gibi çalışmaktadırlar. Sinir ağının sonuna doğru yerleşen katmanlar ise daha çok probleme özgü ayırt edici özelliklerin elde edildiği katmanlardır. Bu sebeplerden dolayı ince ayar işlemi ile yeniden eğitilen evrişimsel sinir ağlarının ilk katmanları eğitim esnasında olabildiğince sabit tutulurken, son katmanların ağırlıkları büyük ölçüde değiştirilmiştir. Böylece son katmanlar probleme özgü özniteliklerin çıkarılmasına uygun hale getirilmiştir. Eğitim esnasında değiştirilen son katmanların, yeni ağırlık değerlerini daha hızlı öğrenmesi sağlanmıştır.

Kişiyi yeniden tanıma problemi için ince ayar işlemine tabi tutulan evrişimsel sinir ağlarının, eğitim esnasında doğruluk-kayıp grafikleri incelenerek yeterli başarımın elde edildiği görüldü. Bu aşamadan sonra sinir ağları, bir insanın dış görünüşü için gerekli ayırt edici özellikleri çıkarabilecek kapasiteye ulaşmıştır. Kişi resimleri baş, gövde ve ayaklar olmak üzere üçe bölündükten sonra ince ayar işlemi yapılarak insan vücudunun her bölgesi için ayrı ayrı evrişimsel sinir ağları oluşturulmuştur. Daha sonra öznitelik çıkarımı aşamasında her vücut bölgesi için ilgili sinir ağı kullanılmıştır. Kişinin nihai temsili ise bu özniteliklerin birleştirilmesiyle elde edilmiştir.

Bu çalışmada, evrişimsel sinir ağlarına uygulanan ince ayar aşamasından sonra iki farklı yöntem izlenmiş ve sonuçları ayrı ayrı raporlanmıştır. Birinci yöntem ikili metrik öğrenme yöntemleri diğeri ise öznitelikler arası doğrudan benzerlik ölçümü işlemidir. İlk olarak, ince ayar işlemi esnasında kullanılan eğitim kümelerinin, ince ayar yapıldığı sinir ağı üzerinden öznitelikleri çıkarılmıştır. Hazırlanan bu öznitelikler ikili bir şekilde positif ve negatif olmak üzere etiketlenmişlerdir. Aynı bireye ait iki resim pozitif etiketlenirken farklı kişilere ait iki resim negatif olarak etiketlenmiştir. Bu şekilde belirli bir sayıda ikili örnek hazırlandıktan sonra, eğitim kümesi metrik öğrenme algoritmalarına sokulmuştur. Metrik öğrenme yöntemlerinin amacı aynı kişiye ait öznitelikler karışılaştırıldığı zaman benzerliğin yüksek, farklı kişiye ait öznitelikler karşılaştırıldığı zaman ise benzerliğin düşük olduğu bir öznitelik uzayına geçişin yapılmasını sağlayan, dönüşüm matrisinin elde edilmesidir. Elde edilen dönüşüm matrisi kullanılarak test kümesindeki kişiler arası benzerlik ölçümü, bu uzaya geçildikten sonra yapılmaktadır. Test kümesindeki kişiler içinde sinir ağlarından öznitelikler elde edilmekte ve elde edilen dönüşüm matrisi ile benzerlik ölçümü yapılmaktadır.

Uyguladığımız diğer bir yöntem ise test kümesindeki kişiler arasında doğrudan benzerlik ölçümünün yapılarak eşleşme oranlarınının elde edilmesidir. Bunun için

öncelikle ince ayar yapılmış sinir ağlarından test kümesindeki kişilerin her bir vücut bölümleri için öznitelik çıkarımları yapıldı. Daha sonra bu öznitelikler birleştirilerek kişilere ait nihai gösterimler elde edildi. Sorgu ve galeri kümesindeki olarak ikiye ayrılan test kümesindeki kişilerin özniteliklerininin benzerlik ölçümü esnasında kosinüs uzaklığı kullanılmıştır. Sonuçlar herbir bölge ve nihai gösterim için ayrı ayrı 10 kez yapıldıktan sonra ortalama değerler yine ayrı ayrı raporlanmıştır.

Eğitim için farklı, test için farklı kümelerin kullanıldığı bu çalışmada doğrudan uzaklık ölçümü sonuçlarında en iyi olarak Rank-1 değerinde %32 eşleşme oranı elde edilmiştir. Bu oran belirtilen senaryoya sahip kişiyi yeniden tanıma problemi için önerilen sonuçlar arasında oldukça yüksek bir değerdedir. Elde edilen sonuçlar diğer yöntemler ile karşılaştırılmıştır.

1. INTRODUCTION

Nowadays, there is a growing demand in video surveillance systems. Thanks to developments in computer vision technologies, we can see these kind of systems in almost every public places, for example, bus stations, city squares, airports and shopping malls. They have an important role to ensure public security and to examine past events. Generally these systems consist of many cameras and monitored by several attendants. But we know that, monitoring of surveillance videos during whole day is a very labour-intensive, tiring and costly task. It is a well known fact that watching video streams needs a higher level of visual attention than many daily works. Especially wakefulness is incredibly hard and prone to error because of environmental distractions [1].

Today's video surveillance systems not only capture, store, and distribute video streams but also have many smart applications to help people and prevent crimes [1]. They can be used for many security purposes in our daily life. For example; asset protection, suspicious people tracking, motion detection, abandoned object detection, face recognition etc. Person re-identification is also one of them and it is very suitable scenario for video surveillance systems. Generally, these kind of applications generate different types of alarms to engage officer's attention for a certain point.

In computer vision and image processing world, person re-identification is considered as one of the most difficult and challenging problems. We can define this problem as the task of assigning the same identifier to all images that are belong to same person [2]. Stated in other words, it aims to answer the question "Where have I seen this person before?" [2]. There are several issues in the person re-identification problem. We can classify them into two main categories: inter-camera and intra-camera [3]. Illumination changes for different scenes, different camera angles, pose changes, similarity of clothing and blurred images can be categorized as inter-camera issues. Background illumination changes, occlusion in frames and low resolution can be categorized as intra-camera issues. All of them make this problem more difficult. Also

conventional video surveillance systems do not provide high enough image resolution most of the time. So, it is hard to use iris or face recognition algorithms. At this point, we need a different kind of solution to distinguish people from each other and recognize them.

Over the last decade, many methods have been proposed by researchers for person re-identification problem. Despite all these methods, there is no indepth solution reported in the literature [2]. Because, this problem contains many parameters and can be used with thousand different scenarios. To simplify the problem and solution approaches, researchers have proposed a multidimensional space [2] as illustrated in Figure 1.1.

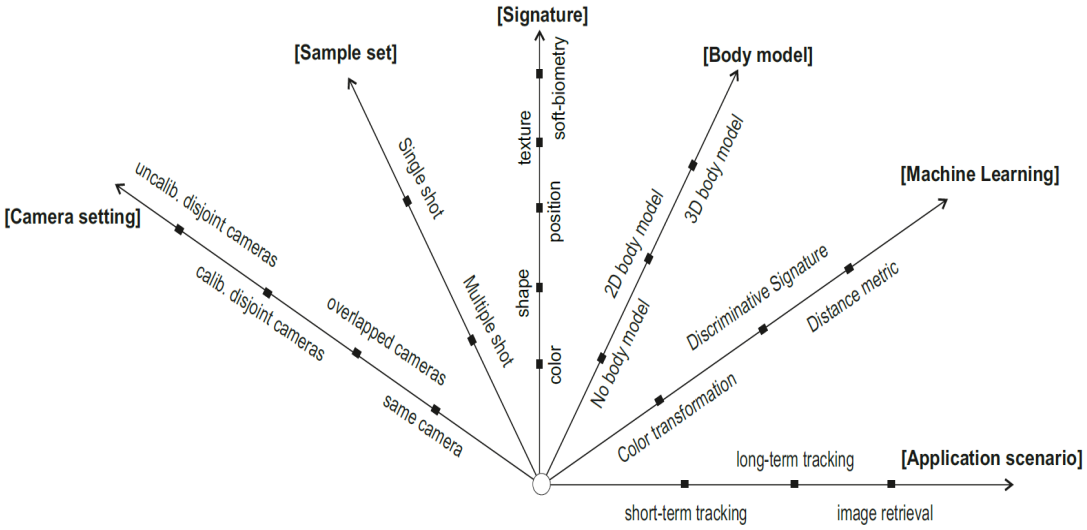


Figure 1.1 : Multidimensional taxonomy for people re-identification algorithms [2].

As we can see in Figure 1.1, the taxonomy of person re-identification problem basically consists of six dimensions [2]. These are camera settings, sample set, signature, body model, machine learning methods, and application scenarios. Furthermore, each dimension has different parameters. In terms of camera settings, we can use the same, overlapped, calibrated disjoint or uncalibrated disjoint cameras in our setup. This selection will define the type of our recorded visual data. Second dimension is the amount of sample set. We can have a single image or a video record for target person. If we have only one image for each person, it is called "Single Shot" or if we have a video record for each person, it is called "Multiple Shot" scenario. The other dimension is signature extraction, which will be used to uniquely represent the each person. Color, shape, position, texture, and soft-biometry features can be used separately or

jointly to create a unique person signature. [2]. The fourth dimension is body model. Generally 2D models are used in person re-identification, but nowadays, there are also new solutions that use 3D models [2]. The machine learning algorithms, which are used during the re-identification process, define the fifth dimension of taxonomy. And the last one specifys the our general person re-identification scenario. For example; if we will use re-identification process with outdoor cameras and long-term tracking purpose, it would be better to use 2D person model in terms of computation process.

When we consider six dimensions which mentioned above, they provide us with over a thousand different combinations of parameters and solutions. But in terms of feasibility, many of them are redundant and not applicable. Despite best efforts, no consistent or conclusive results have been published and person re-identification is still an unresolved problem.

In this work, our person re-identification scenario contains following settings: first of all, our dataset consists of multiple images, which is called as "multiple shot", for each person. These images have been obtained from disjoint cameras. All of them contains 2D human body model. And our comparison methods similar to image retrieval scenario. We also used deep learning based approach which is not mentioned above taxonomy. Lastly, we used different datasets during the training and test phases. This is called "cross dataset person re-identification" scenario in the literature.

Generally, person re-idetification systems take as input video stream, which is processed by person detection algorithms. In person re-identification problem it is assumed that, human detection stage has already been done. After that, manually designed features are extracted from bounding-boxes of persons. To decrease the illumination conditions or geometric transforms, essential transformation processes are applied before the similarity estimation step. Finally, using some metric learning or classification algorithms, the similarity of given two person images are calculated. The typical workflow of a person re-identification system is shown in Figure 1.2 [4].

For the last few years, researchers have been trying to optimize and improve each block of the pipeline which is mentioned in Figure 1.2. Thanks to development of computer and GPU accelerated systems, they can use more complex and robust methods to solve person re-identification problem. Deep learning is one of these new techniques.

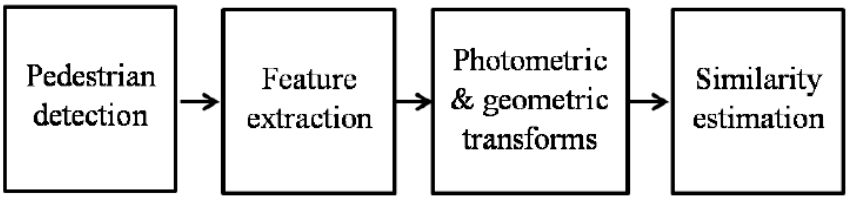


Figure 1.2 : Flow chart of person re-identification problem [4].

It has achieved great success in solving many computer vision problems, including object recognition, image classification, face recognition, scene understanding, object detection and digit recognition. Instead of using hand-crafted features, thanks to deep learning, we can extract more discriminative features automatically from data.

1.1 Purpose of Thesis

The purpose of this thesis is to develop a novel person re-identification system using off-the-shelf deep representations and metric learning methods. Thanks to an increasing amount of data about this problem, deep learning based solutions have recently started to get successful results. It will be stated that doing transfer learning on the off-the-shelf deep representations, could achieve good results in person re-identification problem. Obtained features from deep models also have good discriminative property. It will be shown that these features can be used with metric learning algorithms to match pedestrian images, which are acquired from different cameras. Also to show discriminative power of extracted features, we will use direct similarity estimation using test features.

1.2 Literature Review

In this section, well-known person re-identification algorithms are discussed according to their rankings. As stated in [5], modern techniques have been developed since 2008 in this field. And also, to solve the person re-identification problem, there has been a significiant increase in computer vision research in the past 5 years. When we look at these proposed methods, we can divide them into three categories. First category tries to find distinctive and stable features for describing the human appearance. Second one is metric learning approaches. And the third one is deep learning based methods.

In this section, some of them has been selected according to their number of citations and explained in the following paragraphs.

From the first category, Hamdoun *et al.* (2008) proposed a person re-identification mechanism for multiple camera video surveillance system [6]. The algorithm starts with the extracting frames from short video sequences and then finds some useful interest points. Basically, they collect some interest points from a few images, which are obtained from short video sequences and try to match these points each other. In this algorithm, the usage of image sequences is very important. They used a different version of SURF algorithm instead of SIFT detector or descriptor. This is locally-developed and particularly efficient variant of original SURF algorithm. To match person images, they used a function which implements a different type of Best Bin First (BBF) search. In this work, they needed high quality image sequences obtained from surveillance cameras.

Slawomir Bak *et al.* (2010) proposed a new human body appearance based model. In this method, they extract spatial covariance regions from human body parts [7]. For this purpose, firstly, they used off-the-shelf human and human body parts detectors. The invariant signatures, which are belong to different body regions, are obtained by combining the color and textural information. Then, these signatures are used for the comparing different body parts. To match pedestrian images, they used matching function, which is based on a similarity defined using the covariance matrix distance. One drawback of this method is that, extraction of foreground from pedestrian images is not good enough.

Farenzena *et al.*, in [8], presented an appearance-based method. They select salient parts of human body figure after pre-processing step. While they we doing this, they used principles of symmetry and asymmetry. Firstly, they divide human body into three main regions according to two horizontal axes of asymmetry. Usually these are corresponding to head, body and legs. Then, a vertical axis of human body symmetry is estimated. After that, for each body part, several complementary aspects are detected. According to the distance with respect to these horizontal and vertical axis, each extracted features are weighted to minimize the effects of pose variations. To calculate similarity between extracted features, they tried the match them. Finally, they have used VIPeR dataset to calculate their success rate [9].

D'Angelo and Dugelay, in [10], introduced a color based feature for person re-identification problem. The main purpose of this work is to introduce a new signature based, reliable person re-identification system. To create a unique human signature, they used the color feature. For each person images, they calculate a probabilistic color histogram using a fuzzy classifier. This classifier is trained on an ad-hoc dataset and can be used to match person images which are obtained from different cameras of the network. To match person images, they calculate the distance between probabilistic color histograms of each person images. There is a threshold value to find whether the person images belong to the same person or not. They stated that, their system is very useful for real time scenarios.

Yang Li *et al.*, in [11], proposed a metric learning approach, with an emphasis on the multi-shot scenario. In this method, the procedure starts with the dimension reduction process on image feature vectors. They used random projection matrices for this purpose. Using these matrices, the vectors are projected to their new low dimensional sub-spaces. Then, random forest classifiers are trained based on pairwise constraints in these spaces. Finally, using multi-shot appearances of persons, some of them are selected and used for matching person images.

The second category is metric learning approaches. Weinberger *et al.*, in [12], proposed a new distance metric based method, which is called LMNN and the aim is to learn Mahanalobis distance metric for k-nearest neighbour classification. Koestiger *et al.* [13] proposed a new distance metric learning method, which is called KISSME, with using some constraints based on a statistical inference perspective. Li *et al.* [14] proposed a new joint model for person verification. This model is a combination of distance metric and locally adaptive thresholding rule and called as "Locally-Adaptive Decision Functions" method. Davis *et al.* [15] presented an approach called ITML to formulate the learning of Mahalanobis distance function for person re-identification problem. All of these metric learning methods are very sensitive to parameter selection process. So, it is hard to apply them into real time applications.

The other category is deep learning based approaches. Firstly, Yi *et al.* [16] proposed a deep learning based method, which is called Deep Metric Learning (DML). This method uses "siamese" neural networks to learn useful metrics. This method can jointly learn several features such as color feature, texture feature and distance metric.

The network consists of two symmetric sub-networks and they are connected by a cosine layer. Every subnetwork has two convolutional layers and a fully connected layer. The method tested on cross person re-identification dataset, which is more suitable for real world applications. In this experiment, they had used different sets for training and testing steps.

Wei Li *et al.*, in [4], proposed a filter pairing neural network (FPNN). The method can jointly handle some different problems like geometric transforms, occlusions, and background clutter. Instead of handcrafted features, their method can automatically learn the optimal features for person re-identification task. So, this model can handle complex photometric and geometric transforms by itself. In this work, they also have built the largest benchmark re-identification dataset.

Ejaz Ahmed *et al.*, in [17], proposed a new convolutional neural network based algorithm for person re-identification problem. The network can learn both powerful features and corresponding similarity metrics at the same time. When we give the pair of person images to the network, it can identify directly if these images belong to same person or not. Their results are very impressive both large and small datasets.

1.3 Hypothesis

Person re-identification is still an unsolved problem and researchers have been trying to find robust solutions. Nowadays, deep learning based solutions have achieved good results on this problem. In deep learning approaches, success is directly proportional with the amount of training data. There are several datasets for person re-identification problem but none of them is large enough to use convolutional neural networks directly. So, an alternative would be using transfer learning approach on deep models to produce good results.

Transfer learning is a sub field of machine learning and it tries to use knowledge from different but related domain while solving one problem. [18]. In other words, if you have enough knowledge at some problem space, you can use this experience on other related problems. In the same way, this approach can be applicable for convolutional neural networks. For instance, if you have a CNN model, which was trained with related very large data, you can re-train this model with small dataset,

which is appropriate for your problem. Therefore, this CNN model becomes more suitable for your problem's domain.

Based on this assumption, in this work, using a general convolutional neural network (CNN) model, which was developed for object recognition, a successful system has been introduced for the person re-identification problem. To use this CNN model for the person re-identification problem properly, it has been individually fine-tuned using different body parts of person images. In this work, mainly we focused on transfer learning on AlexNet [19] model, but also VGG-16 [20] and GoogleNet [21] models have been used. It will be shown that, employing transfer learning on pre-trained convolutional neural network models can be useful for person re-identification problem.

After training stage, for feature extraction, we used different layers of the CNN models. Then, we used cosine similarity metric and some other metric learning methods to calculate the similarity between extracted features. CUHK03 [4] and Market-1501 [22] datasets were used as the training sets and the proposed method has been tested on VIPeR [9], PRID 2011 [23] and iLIDS-VID [24] datasets. Superior results have been obtained with the proposed method compared to the state-of-the-art methods in the field.

2. AN OVERVIEW OF CONVOLUTIONAL NEURAL NETWORKS

In this chapter, a brief introduction of deep learning and convolutional neural networks is given. Nowadays, convolutional neural networks (CNNs) and deep learning based solutions provide the best results in many problems. They can be applied in different fields such as image recognition, speech recognition, and natural language processing. Deep learning is a branch of machine learning and its aim is to create higher level representations of the data through the use of multiple layers of non linear operations [25]. These higher level representations are much more useful for classification and recognition then the basic or raw features. There are many deep architectures in the literature and CNNs are one of them. A CNN consists of different number of convolutional and fully connected layers. It also uses tied weights and pooling layers. The general structure of the deep convolutional neural network was introduced in 1998 by LeCun [26]. In Figure 2.1, we can see the complete implementation of the LeNet-5.

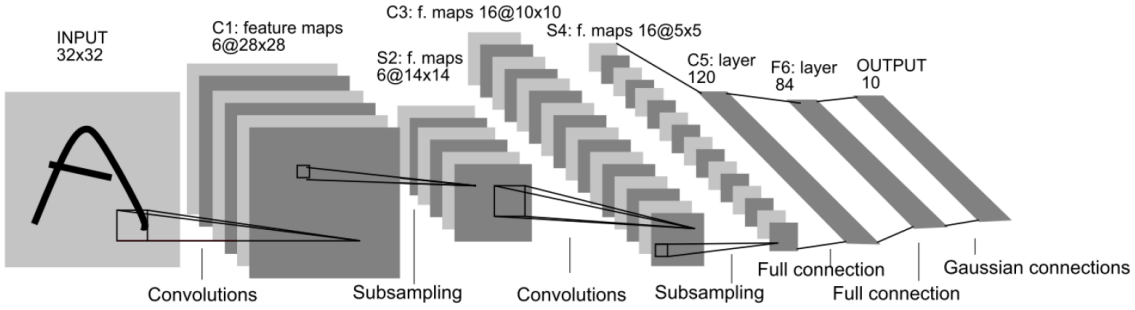


Figure 2.1 : The complete implementation of the LeNet-5 [26].

2.1 Components

A convolutional neural network generally consists of specific set of components. These components are customized according to the problem. In this section, the most important components will be summarized, their role in the neural networks and the advantages and disadvantages of different components will be discussed.

2.1.1 Convolutional layer

The convolutional layer is the main layer of a CNN. This layer consists of a set of learnable filters (kernels). Each filter is responsible for the convolution operation during the forward pass. After that product operation, convolutional layer produces several feature maps depending on its size and dimensions. These maps are stored in the layer and belongs to its filter. The network learns filters automatically when they see some specific type of feature at some spatial position in the input. In Figure 2.2, blocks of the convolutional layer ready to perform dot product with their responsible region. Each filter is convolved with the input data and different feature maps are produced as shown in Figure 2.2.

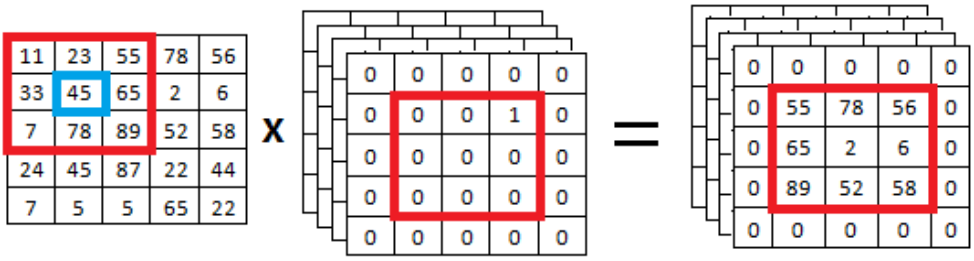


Figure 2.2 : Convolutional layer.

2.1.2 Activation functions

In neural networks, the activation function's aim is transform the input value to the output value of the neuron [27]. Generally, sigmoidal activation function is used as the activation function in neural networks. A sigmoid function produces a curve with an "S" shape and it includes the sigmoid and hyperbolic tangent (tanh) functions. Sigmoid functions have minimum and maximum values. These values cause saturated neurons in the higher layers of the neural network. These saturated neurons determine the depth of the neural network [27].

The other important activation function is rectifier linear unit (ReLU) function. This function is different from the sigmoid function. We can say that, this is a max function as in Equation (2.1). It is bounded by only minimum value which is zero and can represents any non negative real value. The function also has good sparsity properties

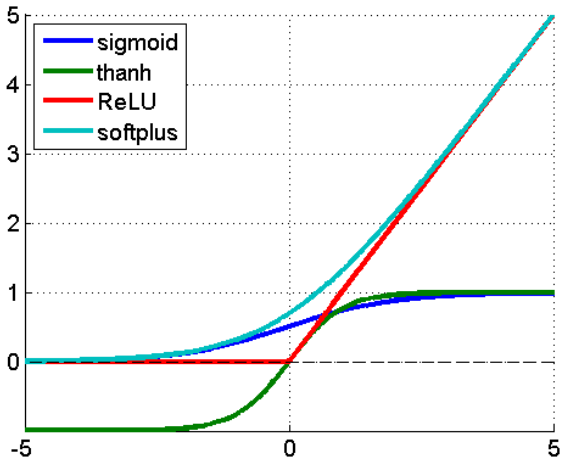


Figure 2.3 : The most used activation functions in neural networks [28].

due to having a real zero activation value [27].

$$rect(x) = max(0,x) \tag{2.1}$$

In Figure 2.3, we can see the most used activation functions, which are sigmoidal, thanh, ReLU and softplus [28], in convolutional neural networks.

2.1.3 Pooling

Pooling is another important concept of CNNs. It is a kind of dimension reduction technique in CNNs. Its goal is to throw away unnecessary information and only preserve the most critical information. In this way, it can also control overfitting. Moreover, to make the detected features invariant from the location in the input image, pooling is used. Generally, this layer is periodically inserted between successive convolutional layers. Basically, there are two kind of pooling functions, which are called as max pooling and average pooling.

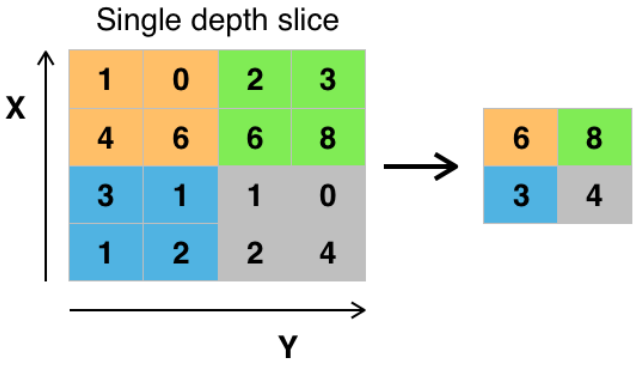


Figure 2.4 : Max pooling with a 2x2 filter and stride = 2 [29].

Max-pooling operation takes the input data and returns the maximum value as shown in Figure 2.4. Generally, the pooling layers have small sized filters such as 2x2. Different from max-pooling, average-pooling returns the average value of input data. There are some disadvantages of using these layers in the network. In terms of average pooling, it can minimize the activation value because of negative and low values. On the other hand, when we use max pooling, we can lost the useful and discriminative information in that layer. Because, it ignores the low values and this is likely to cause over-fitting the dataset quite quickly [27].

2.1.4 Dropout-DropConnect

In machine learning based applications, one of the most common tasks is to fit a model to the traning data. This task is important, because the success rate of our prediction mechanism is highly related to this process. But, if our model is overly complex or not suitable to the problem, overfitting may occur. A model that overfits will generally have poor prediction performance. Additionally, if the model begins to memorize the training data rather than learning it, overfitting is inevitable.

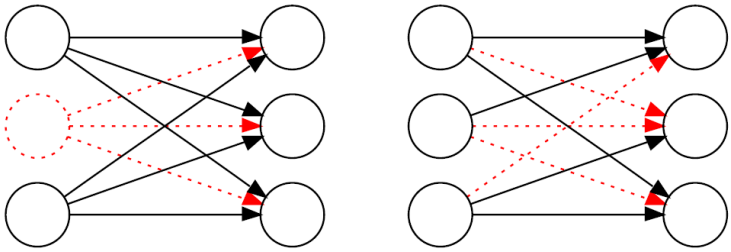


Figure 2.5 : Left: Dropout, right: DropConnect [27].

Dropout is one of the methods that are used to prevent overfitting. It works by randomly omitting half of the neurons for each training case. So, network size is reduced by this operation at the training stage. This makes the neurons more adaptive and less restricted to the existing architecture of the neural network. Using this operation in the networks, gives us more generalized results.

DropConnect also provides similar functionality to dropout: instead of randomly omitting half of the feature detectors it randomly omits the weights. The weights are omitted by setting the weights to zero.

In Figure 2.5, the left side of the figure represents the dropout procedure. Randomly selected neuron, which is shown red-dotted circle, is omitted. And the right side of the figure, we can see dropconnect function. This time, randomly selected weights are ommitted. These are very important functions to avoid overfitting.

2.1.5 Fully connected layer

After different number of convolutional and max pooling layers, fully connected layer(s) is/are located in the network. In this layer, each neuron is connected to the previous layer through its neurons and each connection has it is own weight.

2.1.6 Output layer

The output layer gives probability distrubiton over the output classes. In this layer there are also several loss functions that can be used. Softmax, sigmoid, and euclidean are some of them. Softmax is the most common loss function. It is a linear classifier, as it uses the log probability distribution.

2.2 Training a Convolutional Neural Network

Convolutional neural networks are trained based on training data like the almost every machine learning problem. There are many different algorithms for training neural networks. Generally, they are trained through backpropagation algorithm. In one feature map of a convolutional layer, all neurons share same weights and bias. Because of this, a convolutional layer has fewer parameters then a fully interconnected multilayer perceptron layer and it leads to reduction of the gap $e_{test} - e_{train}$ [30]. Also in this work, to train our networks, we used backpropagation algorithm. In this section, this algorithm will be described briefly.

2.2.1 Backpropagation algorithm

Generally, to learn the weights of a multilayer neural network, the backpropagation algorithm has been used. To adapt the weights, it propagates the errors back through the feedforward architecture. Training a neural netwrok with this algorithm is composed of two simple steps: the feedforward and the backpropagation step [31]. Figure 2.6 will be used to explain these algorithm's steps.

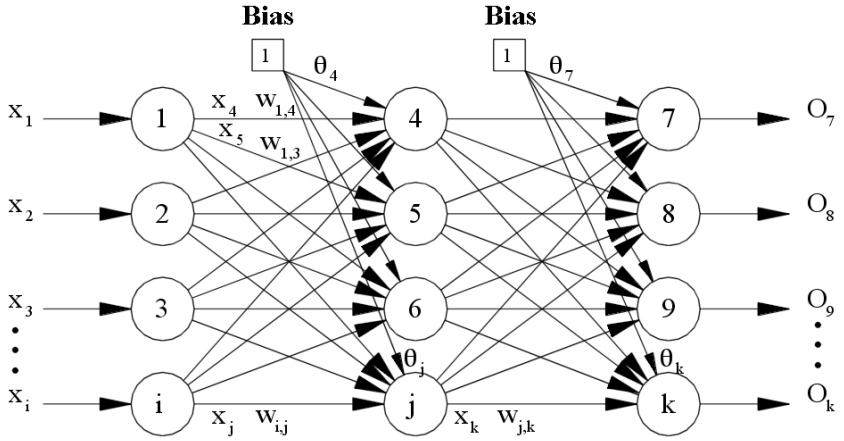


Figure 2.6 : An example neural network with one hidden layer [32].

2.2.1.1 Feedforward

Let's consider a node in a back-propagation network. It has multiple inputs and produces an output value. Also each node has a bias value, θ. The activation is a summation of weighted inputs and the bias value. Let's refer to our node of interest as j, nodes in the preceding layer with values of i, and nodes in the succeeding layer with values of k. The activation of our node j is then;

$$Net_j = \sum(w_{ij}x_j) + \theta_j \tag{2.2}$$

Equation (2.2) is used to calculate the aggregate input to the neuron. The output of our node j is passed onto an appropriate activation function, to decide whether a neuron should ignore. After the activation function process, the resulting value determines the neuron's output. This value will be used as a input value in the next layer. A typical activation function used is the sigmoid function (as shown in Figure 2.3).

$$O_j = f(Net_j) = \frac{1}{(1+e^{-Net_j})} \tag{2.3}$$

Feedforward step is done after all the node values has been passed through the activation funcitons.

2.2.1.2 Backpropagation

In the backpropagation step, a classification error is computed and propagated back through the neural network. The weights are updated based on the error. Backpropagation is the method used to achieve gradient descent in neural networks. So, whatever transfer function we use, we need to know how to compute its first derivative. When we take the first derivative of sigmoid function;

$$\frac{df(Net_j)}{dNet_j} = f(Net_j)(1 - f(Net_j)) \tag{2.4}$$

After that, we need to consider errors and weight adjustment. δ is used for a node error estimate. It is proportional to the first derivative of the node's activation and an error term it receives. There are two formulations for the received error term, one for output nodes and one for hidden nodes. If the actual activation value of the output node, k, is O_k, and the expected target output for node k is t_k, the difference between the actual output and the expected output is given by;

$$\Delta_k = t_k - O_k \tag{2.5}$$

The error signal for node k in the output layer can be calculated as;

$$\delta_k = \Delta_k O_k(1 - O_k) \tag{2.6}$$

or

$$\delta_k = (t_k - O_k)O_k(1 - O_k) \tag{2.7}$$

where $O_k(1 - O_k)$ term is the derivative of the sigmoid function. The formulas used to modify the weight, w_{jk} , between the output node, k, and the node, j is:

$$\Delta_{w_{jk}} = L_r \times \delta_k \times x_k \tag{2.8}$$

$$w_{jk} = w_{jk} + \Delta_{w_{jk}} \tag{2.9}$$

where $\Delta_{w_{jk}}$ is the change in the weight between nodes j and k, L_r is the learning rate. The learning rate indicates the change in weights. If it is too low, the network will learn very slowly. In Equation (2.8), the x_k is the input value to the node k, and is the same value as the output from node j.

To calculate the error value for our hidden node j, as follows:

$$\delta_j = \frac{df(Net_j)}{dNet_j} \times \sum_k (\delta_k \times w_{jk}) \tag{2.10}$$

Using this, we can calculate how to adjust the weights going to the preceding layer of nodes;

$$\Delta w_{ij} = L \times O_i \times \delta_j \tag{2.11}$$

$$w_{ij} = w_{ij} + \Delta_{w_{ij}} \tag{2.12}$$

Δw_{ij} here represents the changes in weights. After all these calculations, the changes are applied to the whole network.

2.3 Off-the-Shelf Deep Neural Network Architectures

In this section well-known deep neural networks are explained in detail. There are several trained convolutional neural networks in the literature. In this thesis, AlexNet [19], VGG-16 [20] and GoogleNet [21] models have been used for transfer learning. These networks are quite different from each other and also each of them is specialized to specific problems.

2.3.1 AlexNet

AlexNet is a convolutional neural network architecture, which was developed for object recognition by Krizhevsky *et al.* [19]. This neural network, trained on the 1.2 million image (ILSVRC 2012 ImageNet [33]) dataset and achieved the state-of-the-art performance [19]. In [19] Krizhevsky *et al.* states that simple recognition tasks can be solved quite well with small sized datasets. But in real world conditions, to achieve good recognition rates, it is necessary to use much larger training sets. So, they develop a new neural network model with a large learning capacity. The architecture of their network is shown in Figure 2.7.

The network contains eight layers in total; the first five of them are convolutional and the remaining three are fully-connected. The 1000-way softmax layer, which produces a distribution over the 1000 class labels, feeds the output of the last fully-connected layer. As shown in Figure 2.7, the kernels of the second, fourth, and fifth convolutional layers are connected only to those kernel maps in the previous layer. Unlike them, the third kernel is connected to all kernel maps in the second layer. After every

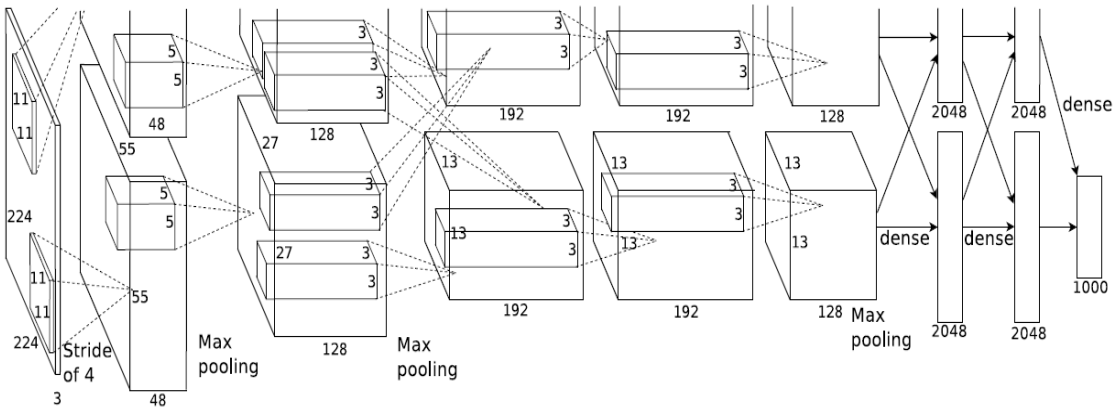


Figure 2.7 : AlexNet architecture [19].

convolutional and fully-connected layer, the ReLU function is applied throughout the network. Also the number of kernels and their sizes are shown in Figure 2.7. The first convolutional layer filters take 224x224x3 input image with 96 kernels of size 11x11x3 with a stride of 4 pixel [19]. The second convolutional layer takes as input the output of the first convolutional layer and filters it with 256 kernels of size 5x5x48 [19]. Without any intervening pooling or normalization layers, the third, fourth, and fifth convolutional layers are connected to one another. These layers have different number and size of kernels from each other. Their kernels and sizes are as follows: the third convolutional layer has 384 kernels of size 3x3x256 and it connected to the outputs of the second convolutional layer. The fourth one has 384 kernels of size 3x3x192, and the fifth convolutional layer has 256 kernels of size 3x3x192. Finally, each fully-connected layers have 4096 neurons [19].

AlexNet [19] neural network architecture has 60 million parameters and contains 650.000 neurons. The last layer has probability distrubition over 1000 different classes. When we consider that many parameters, overfitting becomes a serious problem to be solved. The authors have mentioned this problem in their study and explained how to solve it. They used two primary ways to combat overfitting; first one is data augmentation and the other one is dropout technique. Additionaly, they realized that depth of their neural network and how much data they used at training stage is very important for classification performance.

2.3.2 VGG-16

In [20], Simonyan and Zisserman have introduced very deep convolutional neural network for image recognition. Their main contribution is increasing depth of the network and using very small (3x3) convolutional filters to improve their classification accuracy. They also show that their representations generalize well to other datasets. This network is trained on the ILSVRC 2012 ImageNet [33] dataset like AlexNet [19] architecture, which is described in Section 2.3.1. Their results are ranked in the second place at ImageNet ILSVRC-2014 benchmark. The network has two different configurations, which are 16-layer and 19-layer. In this work, we used their 16-layer version. The 16-layer version of the network is shown in Figure 2.8.

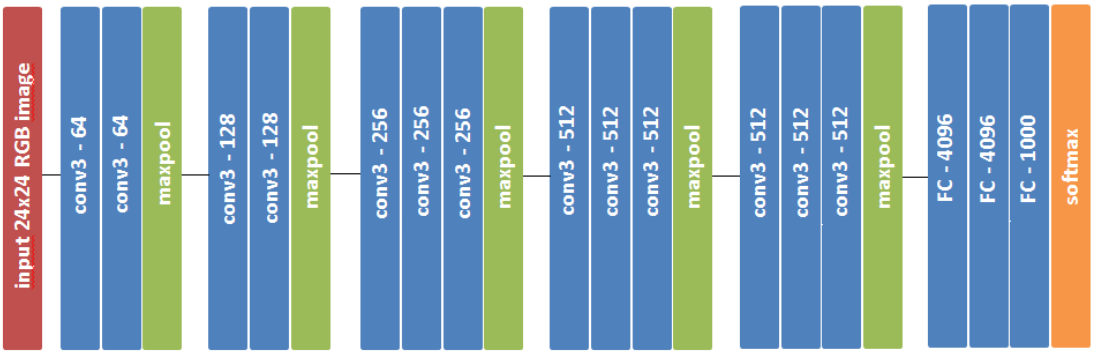


Figure 2.8 : VGG-16 network architecture [20].

The detailed description of the network is as follows. The network starts with the input layer, which uses 24x24x3 RGB image. In 16-layered version of VGG model, there are 13 convolutional layers with very small size (3x3) kernels. After some of the convolutional layers, pooling is carried out by five max-pooling layers. And max-pooling operation is performed over a 2x2 pixel window, with stride 2. After several convolutional layers which have a different depth, there are 3 fully connected layers. The first two have 4096 nodes each and the third contains 1000 nodes. Because the third one performs 1000-way classification (one for each class). The final layer is the softmax layer. Moreover, all hidden layers are equipped with the rectification (ReLU) non-linearity. Finally, in this network the results confirm the importance of depth in visual representations [20].

2.3.3 GoogLeNet

The last convolutional neural network that will be described in this work is GoogLeNet [21]. It was developed for image recognition task and uses 12x fewer parameters than the AlexNet [19], while being more accurate. In this work, two main drawbacks have been mentioned by the authors. We know that from the previous networks, the most straightforward way of improving the performance of deep neural networks is by increasing their size [20]. This can be done by two different ways. First one is increasing the depth of the network and the second one is increasing its width, which means the number of units at each level. Actually this is an easy and safe way of training higher quality models. But this way has also some disadvantages. Firstly, bigger size network means a larger number of parameters, which makes the enlarged network more prone to overfitting. Secondly, when the network size is increased, usage of computational resources are increased as well. To overcome these problems, the authors have introduced a new kind of layer, which is called "sparsely connected layer" [21].

GoogLeNet [21] convolutional neural network with all the layers and filter sizes is as shown in Figure 2.9. The network starts with input field, which takes 224x224 RGB color image. For dimension reduction 1x1 convolutional layers have been used with rectified linear activation. Also there are several dropout layers in the network and they have %70 ratio of dropped outputs. As a classifier, there is a linear layer with softmax loss. When we count only the layers, the network is 22 layers deep. The usage of average pooling before the classifier is also different in this network, because they use an extra linear layer.

Figure 2.9 : GoogLeNet network architecture [21].

3. TRANSFER LEARNING ON CONVOLUTIONAL NEURAL NETWORKS

In this section, transfer learning approach and its implementation on convolutional neural networks are explained. Firstly, "what is transfer learning?" and "how this method can be applied in CNNs?" questions will be answered, then, our fine-tuned results on the well-known networks will be provided in order to show proposed method's learning capacity.

3.1 Transfer Learning in a Nutshell

As described in [34], transfer learning is the improvement of learning in a new problem using the knowledge from a related task, which has already been learned. Generally, machine learning algorithms are designed for the specific problems. They use training and test data from the same domain space. Transfer learning attempts to change this approach by developing algorithms. The aim is to be able to use training and testing data which are come from different domains or distributions. Actually this is very important, because in many real world applications, it is hard to collect required training data and rebuild the models. If the knowledge transfer is done successfully, we can get rid of the expensive data labeling efforts and also we can improve the performance of our learning capacity.

In Figure 3.1, we can see the difference between the traditional and transfer learning techniques and their learning processes [35]. The first one tries to learn each task from scratch. The second one is different and it tries to transfer some knowledge from previous tasks to a target task. In the second one, the training data that we have is less than the first one. So, we use the knowledge from previous tasks instead of training data.

There are several settings used in transfer learning. These are inductive transfer learning, transductive transfer learning, and unsupervised transfer learning, based on difference between domain and task space and whether the training data is labeled or not [35]. In the inductive transfer learning setting, the target and source tasks are

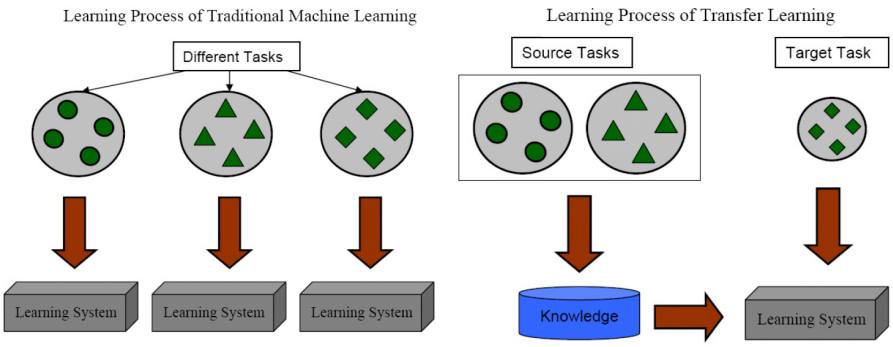


Figure 3.1 : Traditional learning and transfer learning [35].

different from each other and it is not important whether they are come from same or different domains [35]. In the transductive transfer learning setting, the source and target tasks are the same but their domains are different. And lastly, in the unsupervised transfer learning setting, the target task is different from the source task but related as can be.

In this thesis, inductive transfer learning setting is used. Because, our neural networks have been trained for image recognition task before. To use these networks in person re-identification problem, we need to transfer this knowledge from image recognition to person re-identifiaction problem. At this point, the target task, which is person re-identification, is different from the source task and also their domains are different. In terms of CNNs, the transfer learning means not only re-training of convolutional neural network with the new training data but also fine-tuning the weights of the pretrained network while backpropagation process still goes on.

3.2 Fine-Tuning the Convolutional Neural Networks

For many problems, it is really hard to find sufficient amount of training data. Person re-identification is one of them. As previously mentioned in Section 2.3, in order to avoid overfitting, convolutional neural networks need a large amount of training data. So, instead of train an entire network from scratch, pre-trained networks have been fine-tuned with the new training sets. This technique allows convolutional neural networks to be successfully applied to the problems with small training sets.

Fine tuning is one of the transfer learning strategies that used in convolutional neural networks. This is not only re-train the network with the new dataset, but also fine-tuning the weights of pretrained network while backpropagation process still goes

on. We can fine-tune all the layers or we can keep some of them and only fine-tune some layers of the network. It is generally accepted that to avoid overfitting, earlier layers are kept as the same, only some higher layers are fine-tuned. The earlier layers represent more generic features (e.g. edge features, blob features), but higher layers contain more specific features which are about the problem space.

The other important issue involved is how to decide to use fine-tuning operation in our problems. Basically there are two important parameters: the size of the new dataset and its similarity to the original dataset [36]. In person re-identification problem, the training dataset is small and quite different from the original training dataset. This means fine-tuning procedure is a good option for the problem.

3.3 Fine-tuning Results

In this work, AlexNet [19], VGG-16 [20], and GoogLeNet [21] convolutional neural networks are fine-tuned to transfer their knowledge from image recognition to person re-identification domain. Market-1501 [22] and CUHK03 [4] datasets have been used for this purpose. According to our research, currently these are the largest person re-identification datasets. Before the fine-tuning operation, person images are divided into three overlapped parts and each of them is used individually. Additionally, each reference model is fine-tuned with entire person image without dividing three parts. So, four different network models are obtained for one training dataset.

3.3.1 Fine-tuning on AlexNet

As described in Section 2.3.1, AlexNet [19] is a well-known neural network model in image recognition domain. This network has been trained before to recognize 1000 categories of generic objects that are part of the ImageNet hierarchy. So, its weights and learned features belong to a broad domain and its classification function targeted at minimizing error in that domain. To minimize the error, we optimize the network again in more specific domain and replace its classification function. So, the features and parameters of the network have been transferred from the broad domain to the specific one.

To start the fine-tuning procedure, we remove its softmax classifier and initialize a new one with random values. Because, in the original network the classification function

is a softmax classifier, which computes the probability of 1000 classes. In CUHK03 dataset [4] we have 1467 different person and in Market-1501 dataset [22] we have 1501 different person. We can think that, each person represents a different category. So, the new softmax classifier is trained from scratch using the back propagation algorithm with CUHK03 [4] and Market-1501 [22] datasets respectively.

It is very important to set the learning rates of each layer appropriately and it must be done before the fine-tuning operation. The learning rate of new softmax layer has to be large. Because, it has been just initialized with random values. For this reason, we decrease the overall learning rate, but boost the learning rate on the newly introduced layer. So, the new layer can learn its new weights fast, but the other layers' weights change very slowly with the new data. Also we want to preserve the parameters of the earlier layers. After these settings, person images are divided into three parts. For each person part, we train a CNN and extract features of this part as shown in Figure 3.2.

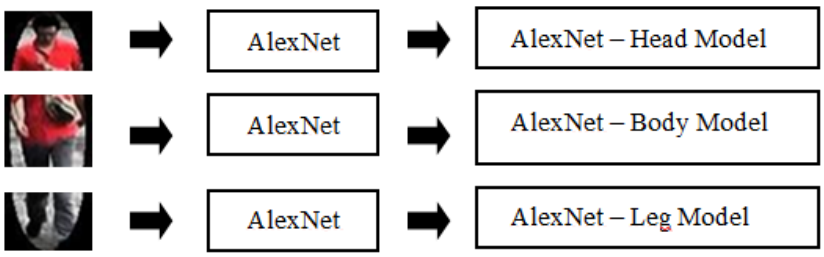


Figure 3.2 : Fine-tuning procedure for each body part.

Additionally, the network has been trained with person images without dividing them as shown in Figure 3.3. Totally four trained networks are obtained for one dataset. We applied this procedure using CUHK03 [4] and Market-1501 [22] datasets seperately. Then, we combined these datasets and trained our AlexNet [19] again. At this time our softmax classifier computes the probability of 2,968 different person.

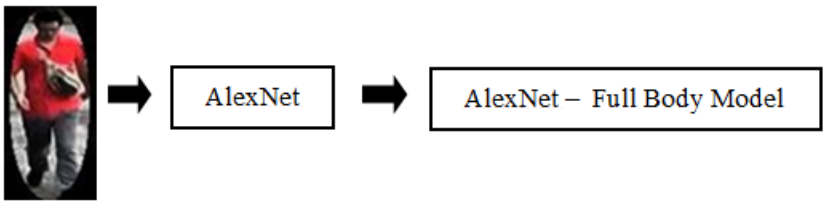


Figure 3.3 : Fine-tuning procedure for full body.

In our experiments, we set the learning rate of the top classification layer to 10, while leaving the learning rate of all the other seven layers to 0.1. As mentioned above, the

idea is that let the new layer learn fast while changing early layer parameters slowly with the new data. We have changed the learning rates of convolutional neural network as shown in Figure 3.4.

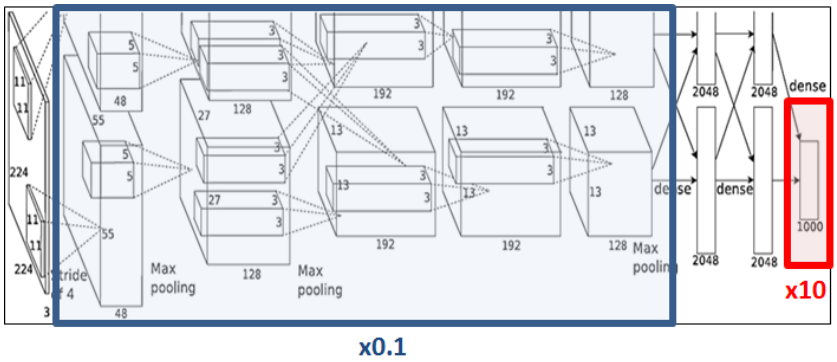


Figure 3.4 : New learning rates for AlexNet [19].

To optimize the network parameters we used stochastic gradient descent (SGD) algorithm and we run the back-propagation algorithm for 50,000 iterations. Each training stage converges in roughly 4-5 hours on NVIDIA GeForce GTX TITAN X GPU (3,072 cores and 12 GB of RAM). For CUHK03 [4] dataset, accuracy and loss functions changes are illustrated in Figure 3.5.

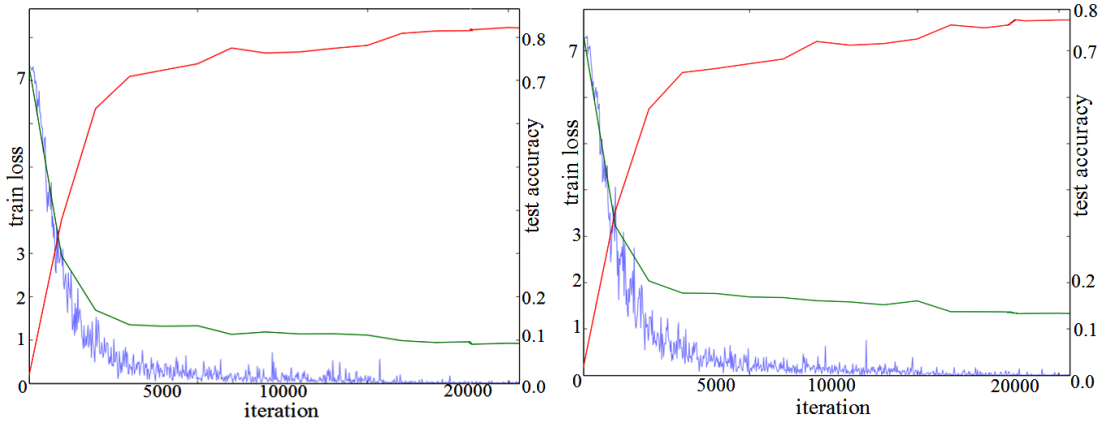
As shown in Figure 3.5, head and full body training results are better than the others. This means head and full-body apperance has more discriminative power. Accuracy results on validation sets are shown in Table 3.1. For validation purpose, we divide the training set into two different sets. We used 70% of them as a traning set and %30 of them as a validation set. The blue lines indicate that loss function change across iteration count. Market-1501 [22] and combination of Market-1501 [22] and CUHK03 [4] dataset's accuracy-loss graphics can be found in Appendix A.1.

Table 3.1 : Fine-tunig accuracy results for AlexNet [19].

DataSet	Head	Body	Leg	Full-Body
CUHK03	%82.6	%76.3	%42.3	%90.3
Market-1501	%72.7	%79.3	%45.7	%89.7
CUHK03 + Market-1501	%76.3	%77.8	%45.6	%89.9

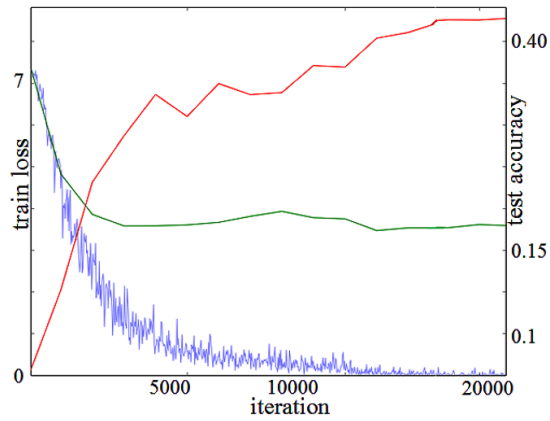
3.3.2 Fine-tuning on VGG-16

Another neural network that we have used for fine-tuning is VGG-16 [20] model. This network is deeper than the AlexNet [19] architecture. It has more convolutional

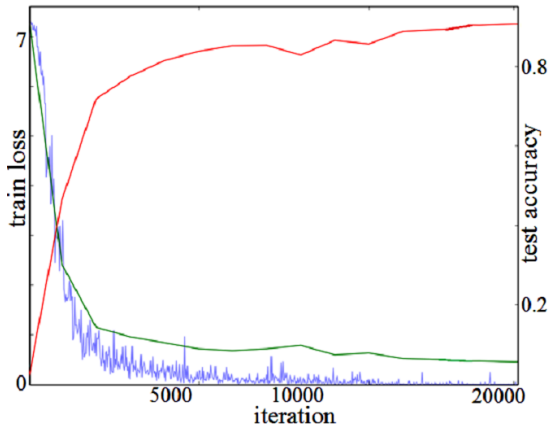


(a) Accuracy-loss graph for head part.

(b) Accuracy-loss graph for body part.



(c) Accuracy-loss graph for legs part.

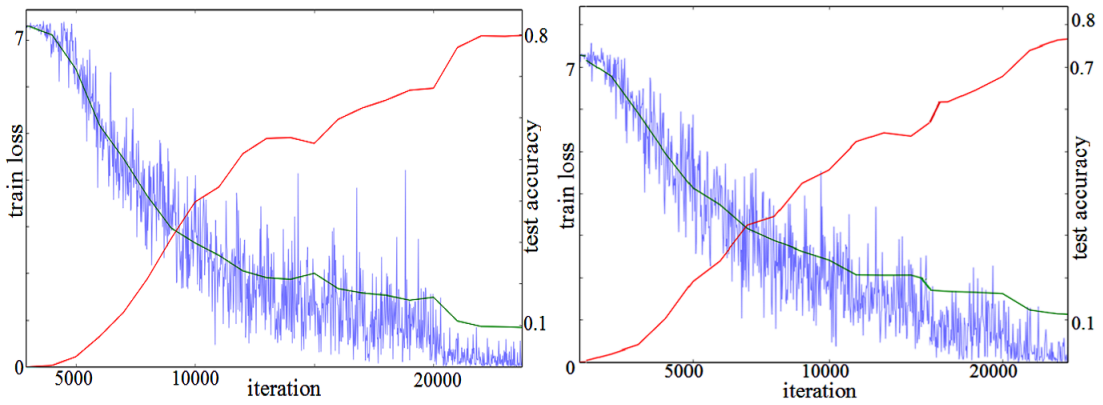


(d) Accuracy-loss graph for full-body.

Figure 3.5 : AlexNet [19] fine-tuning results for CUHK03 [4] dataset.

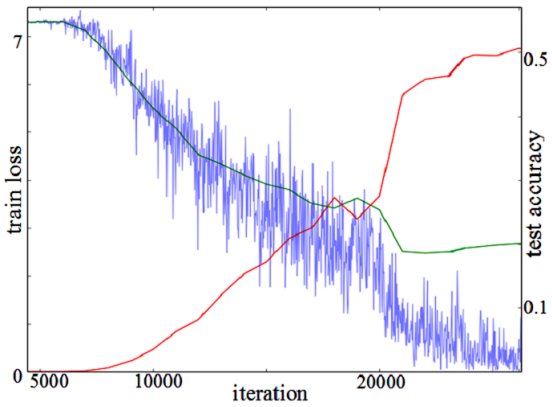
layers and it is also specialized for image recoginiton. Similarly, its softmax classifier computes the probability of 1,000 classes.

At fine-tuning stage, we followed the same procedure, which we used before in AlexNet [19]. First, we changed the softmax layer structure according to our datasets. Then, we increased its learning rate. However, we did not set the rest of the layers' learning rates to zero, they have been just optimized again at a slower pace. In the same way, person images are divided into three parts and then used for fine-tuning operation individually. We run back propagation algorithm for 50,000 iterations. Each training stage converges in roughly 20-21 hours on same GPU settings. The results, obtained on the CUHK03 [4] dataset, are shown in Figure 3.6. The other results that belong to Market-1501 [22] dataset and combination of CUHK03 [4] and Market-1501 [22] dataset, can be found in Appendix A.1.

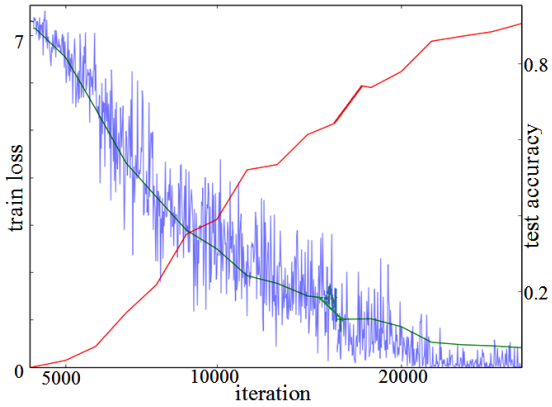


(a) Accuracy-loss graph for head part.

(b) Accuracy-loss graph for body part.



(c) Accuracy-loss graph for leg part.



(d) Accuracy-loss graph for full-body.

Figure 3.6 : VGG-16 [20] fine-tuning results for CUHK03 [4] dataset.

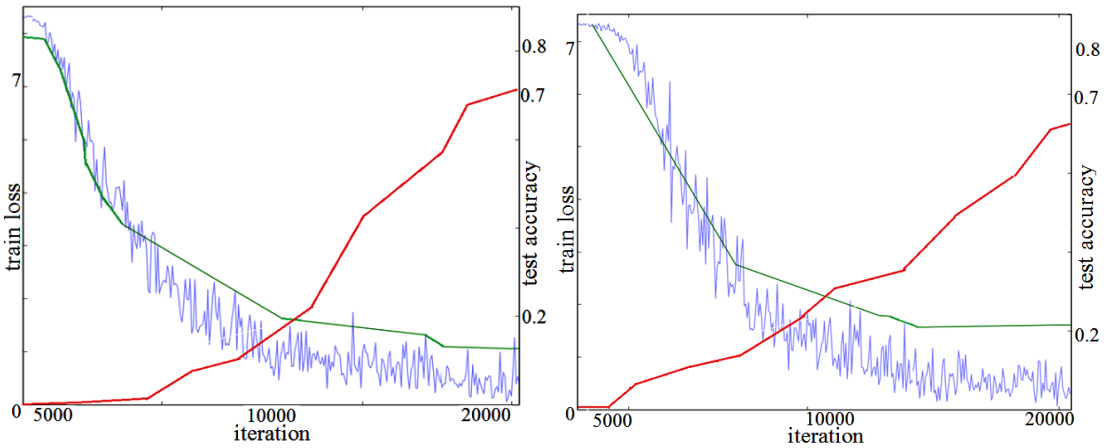
Figure 3.6 shows how the precision of classifying single images improves with more iterations. Network accuracy results on validation data are shown in Table 3.2. We can clearly say that, head and full-body training results are much better than the others.

Table 3.2 : Fine-tuning accuracy results for VGG-16 [20].

DataSet	Head	Body	Leg	Full-Body
CUHK03	%82.1	%77.4	%52.3	%90.3
Market-1501	%68.2	%73.2	%45.1	%88.4
CUHK03 + Market-1501	%62.3	%68.3	%39.7	%84.2

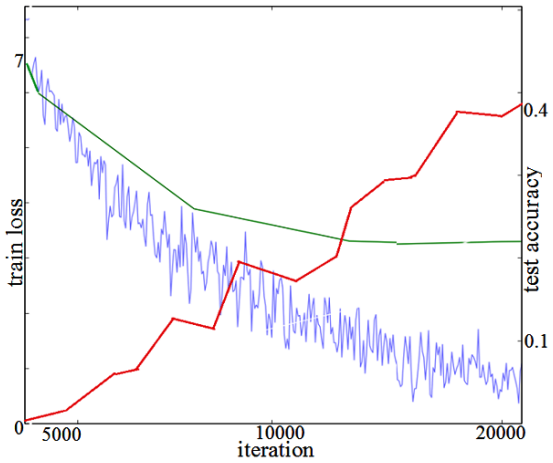
3.3.3 Fine-tuning on GoogleNet

The last neural network that we used for fine-tuning operation is GoogLeNet [21] model. This model has more complexity than the others. It is a 22-layer deep convolutional network, trained on ImageNet data to detect 1,000 different image types. We modified the *"loss3/classifier"* layer and the last layers output up to our datasets. The change of accuracy based on iteration is shown in Figure 3.7.

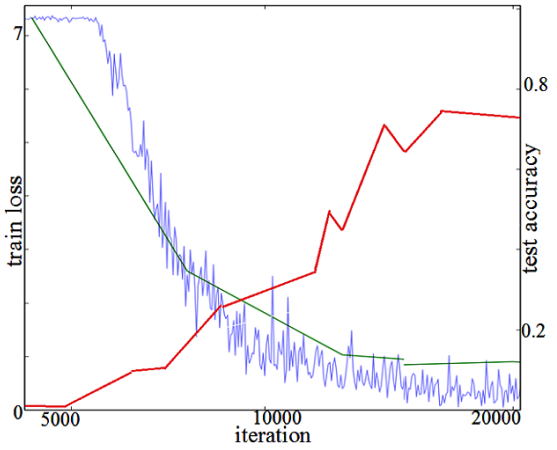


(a) Head accuracy-loss graph.

(b) Body accuracy graph.



(c) Leg accuracy graph.



(d) Full-body accuracy graph.

Figure 3.7 : CUHK03 [4] fine-tuning results for VGG-16 [20].

The network's accuracy on the validation set is shown in Table 3.3. As we can see in the Table 3.3, we have obtained the best result using combination of CUHK03 [4] and Market-1501 [22] datasets.

Table 3.3 : Fine-tune accuracy results for GoogLeNet [21].

DataSet	Head	Body	Leg	Full-Body
CUHK03	%72.1	%68.4	%43.5	%75.4
Market-1501	%68.2	%73.2	%45.1	%66.8
CUHK03 + Market-1501	%62.3	%68.3	%39.7	%78.8

4. FEATURE EXTRACTION AND METRIC LEARNING

In this chapter, feature extraction methods from the fine-tuned networks and the metric learning methods which we used in this work, are explained.

4.1 Feature Extraction from Fine-Tuned Networks

In machine learning, pattern recognition and image processing problems, the training sets can be very large. So before starting to process this amount of data, it is important to reduce its size. There are several dimension reduction techniques in the literature and feature extraction is one of them. It means that, we can reduce the amount of resources required to describe a large set of data [37]. In terms of computer vision, it is a kind of dimensionality reduction technique and represents the images as a compact feature vector. This approach is very useful especially when image sizes are large or we have huge amount of data. There are several feature extraction techniques in the literature, such as Local Binary Patterns (LBP), Histogram of Oriented Gradients (HOG), Speeded Up Robust Features (SURF) and color histograms [38]. Convolutional neural networks can also be used as a good feature extractor.

After fine-tuning procedure, the pre-trained networks are ready to extract discriminative features based on our task. Their convolution layers can be thought as a feature extractors. When we send an image through the network to extract its features, firstly, the convolution operation is applied in every convolutional layer. Then, to reduce the size of the image, the pooling layer eliminates the redundant features and keeps the useful ones. These two layers are the basis of feature extraction part. Afterward, the extracted features are weighted and combined in the fully-connected layer. Generally, a $1 \times D$ feature vector is obtained from these fully-connected layers. Then, these vectors can be used with different classifiers (e.g. SVM classifier).

As mentioned previously in section 3.3.1, each person image is divided into three parts and then we train a CNN with using these body parts. For feature extraction, in the

same way, each person image is divided into three parts and then they had been sent to related CNN individually. Finally, we concatenated these three learned features to get the final person representation. For this procedure, we used different layers of convolutional neural networks.

For AlexNet [19] architecture, FC-6 and FC-7 layers have been used to extract features. These layers produce a 1×4096 dimensional feature vector. That is, for each person part we have a 1×4096 dimensional feature vector. To get the final person representation, we concatenate them, which makes the final feature vector size 1×12288 dimensional. This operation can be summarized as in Figure 4.1

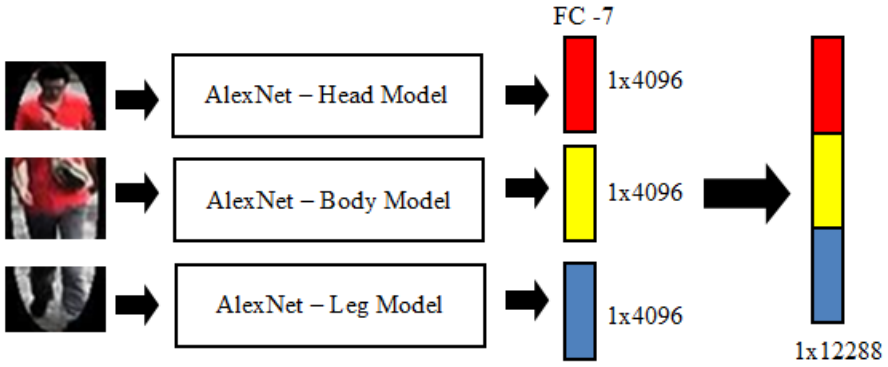


Figure 4.1 : Person representation.

To use in our experiments, we also extract full-body features without dividing person images into parts. This time, fine-tuned full-body model is used to extract features. 1×4096 feature vector has been obtained from the new model as shown in Figure 4.2.

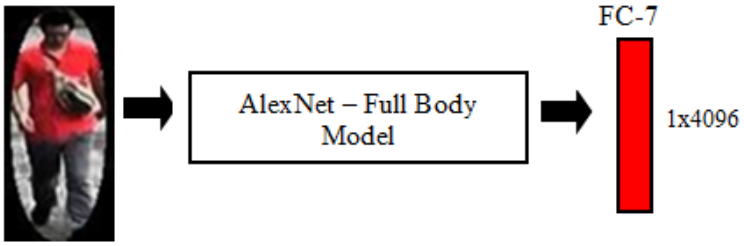


Figure 4.2 : Person representation.

Secondly, in order to extract features, we used fine-tuned VGG-16 [20] convolutional neural network. It has two fully-connected layer at the end of the network. These layers produce 1×4096 dimensional feature vectors. As we did in AlexNet [19] model,

before using these feature vectors, firstly we concatenate them to obtain the final person representation. FC-6 and FC-7 layers have been used for feature extraction process.

Finally, the fine-tuned GoogLeNet [21] network has been used for feature extraction purpose. This network is a bit different from the others. It has many fully connected layers to extract different size of features. We used its "pool5-7x7-s1" layer in order to extract features. This is the last layer before the final softmax in the end, and it contains 1024 elements. Feature vector concatenation procedure was performed as described previously.

4.2 Metric Learning Methods

Metric learning has become a widespreadly used tool in machine learning. It is the task of learning a distance function over sample set [39]. In many problems, it is important to measure how similar or related two sample are. Finding good metric, which describes the dataset, defines the success rate of our classification results.

Metric learning aims to learn a metric from data automatically and using this, project the dataset into a more discriminative space. Generally, pairwise objects which have been labeled as similar or dissimilar, are used in re-identification and verification problems.

In this work, joint metric and diagonal metric learning techniques have been used right after the feature extraction step. High dimensional featue vectors are compressed into low dimensional and discriminative representation using metric learning. This section describes these methods briefly.

4.2.1 Large margin dimensionality reduction

The goal of metric learning is to adapt some pairwise metric function to the any classification problem using the information brought by training examples. We used Mahalanobis distance to learn suitable metric from pairwised person images. In Mahalanobis distance $d_M(\phi_i, \phi_j) = \sqrt{(\phi_i - \phi_j)^T M (\phi_i - \phi_j)}$, the aim is to learn a M matrix which is $M = W^T W$. This matrix projects the high dimensional feature vector to lower dimensional vector. For example the squared Euclidean distance: $d_w^2(\phi_i, \phi_j) = \|W_{\phi_i} - W_{\phi_j}\|_2^2$ of person images i and j is smaller than a learned threshold b, if i and

j belong to the same identity. Otherwise they belong to different people. In [40], these conditions are imposed with a margin at least one as the following constraint: $y_{ij}(b - d_w^2(\phi_i, \phi_j)) > 1$, where y_{ij} is 1 if the person images belong to same person and otherwise it has -1 value. The person image pairs are seperated by margin with at least b learned threshold as seen in Figure 4.3.

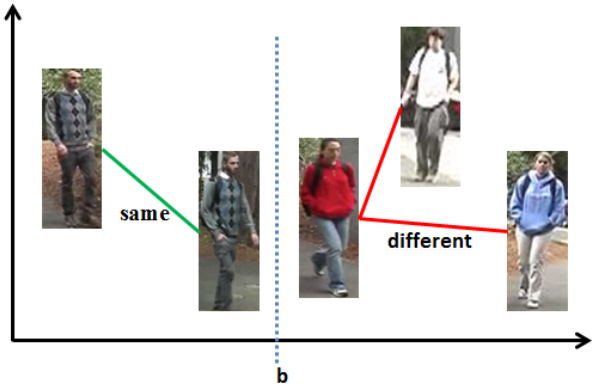


Figure 4.3 : Positive and negative person pairs are seperated with a margin by metric learning.

The Euclidean distance in the lower subspace of m-dimensional can be seen as a low rank Mahalanobis distance in the original n-dimensional space.

$$d_w^2(\phi_i, \phi_j) = \|W_{\phi_i} - W_{\phi_j}\|_2^2 = (\phi_i - \phi_j)^T W^T W(\phi_i - \phi_j) \tag{4.1}$$

4.2.2 Joint metric learning

In [40], a new metric learning method was introduced to measure similarity between face images. It can also be used in pairwise person re-identification problem. This method is similar to low rank Mahalanobis distance, but there is a bit difference. It is corresponding to the difference between low rank Mahalanobis distance ($(\phi_i - \phi_j)^T W^T W(\phi_i - \phi_j)$) and low-rank kernel($\phi_i^T V^T V \phi_j$). Obtained values from these equations are used as distance and similarity score individually. While low-rank Mahalanobis metric gives us distance between pairs image, the value obtained from the inner product gives similarity score. It means small distance equals large similarity score. The optimization equation for joint Bayesian is as shown below;

$$\underset{W,V,b}{argmin} \sum_{i,j} max[1 - y_{ij}(b - ((\phi_i - \phi_j)^T W^T W(\phi_i - \phi_j)) - \phi_i^T V^T V \phi_j), 0] \tag{4.2}$$

To minimize the Equation 4.2, stochastic sub-gradient descent (SGD) method is used. V and W projection matrices are updated every iteration as shown in Equations 4.3 and

4.4 individually.

$$W_{t+1} = \begin{cases} W_t, & \text{if } y_{ij}(b - (d_w^2(\phi_i, \phi_j) - (V\phi_i)^T(V\phi_j)) > 1). \\ W_t - y_{ij}(\gamma d_w^2(\phi_i, \phi_j)(\phi_i - \phi_j)), & \text{otherwise.} \end{cases} \tag{4.3}$$

$$V_{t+1} = \begin{cases} V_t, & \text{if } y_{ij}(b - (d_w^2(\phi_i, \phi_j) - (V\phi_i)^T(V\phi_j)) > 1). \\ V_t + y_{ij}((\gamma V\phi_i)\phi_j^T + (\gamma V\phi_j)\phi_i^T), & \text{otherwise.} \end{cases} \tag{4.4}$$

Depending on the person images belong to the same person or not, y_{ij} value can be 1 or -1. Additionally, learning rate(γ) can be set manually or obtained from training data. Finally, the threshold value (b), which is the seperator for positive and negative pairs, is updated in joint metric learning as in Equation 4.5

$$b = b + y_{ij} * \gamma \tag{4.5}$$

4.2.3 Diagonal metric learning

The other metric learning method that we used in this work is diagonal metric learning. In this technique, the W matrix is assumed to be diagonal. To learn diagonal W matrix, learned feature vector weights and Euclidean distance of re-weighted dimensions are used. The optimization equation for diagonal metric is as shown in Equation 4.6.

$$\underset{u_k \geq 0}{argmin} \sum_{i,j} max[1 - y_{ij}(b - d_u^2(\phi_i - \phi_j)), 0] \tag{4.6}$$

The projection matrix (W) is updated as shown in Equation 4.7;

$$W_{t+1} = \begin{cases} W_t(1 - \gamma\lambda) - (\gamma X_t), & \text{if } W^T X > -1. \\ W_t(1 - \gamma\lambda), & \text{otherwise.} \end{cases} \tag{4.7}$$

In Equation 4.7, gamma and lambda are the learning rates and t is the iteration of sub-gradient stochastic descent. X_t is the difference between squared distance of positive and negative pairs. The X value can be calculated as in Equation 4.8 and gamma is updated in each iteration as shown in Equation 4.9.

$$X = (\phi_1^i - \phi_2^i)^2 - (\phi^x - \phi^y)^2 \tag{4.8}$$

$$\gamma_{t+1} = t\lambda_t \tag{4.9}$$

In Equation 4.8, (ϕ_1^i, ϕ_2^i) represents the feature vectors of positive pairs and (ϕ^x, ϕ^y) represents the feature vectors of negative pairs.

5. CLASSIFICATION

In this chapter, details of the classification algorithm that we used in our experiments are explained. K-nearest neighbors algorithm (KNN) and some distance metrics have been used to calculate similarity of two person images. One of these two images belong to probe set and the other one belong to gallery set.

5.1 K-Nearest Neighbors Algorithm

In machine learning, the k-Nearest Neighbors algorithm is one of the nonparametric methods used for classification. It consists of two major steps. In training step, we have training examples, which are feature vectors in multidimensional space, and their labels. We keep these samples and their labels as they are. In the test (classification) phase, we have an unlabeled vector and we want to find its label. To do this, nearest vectors are used. The k value determine the number of neighbor vectors, which will be considered to find test vector's label. For example, if the k value equals three, an unlabeled vector is classified by assigning the label, which is most frequent among the three training samples that are closest to that query (unlabeled vector) point. If the k is equal to 1, the nearest vector's label is assigned to the test vector directly. To calculate nearest sample point, several distance metrics can be used. An example test phase for k=3 can be seen in Figure 5.1

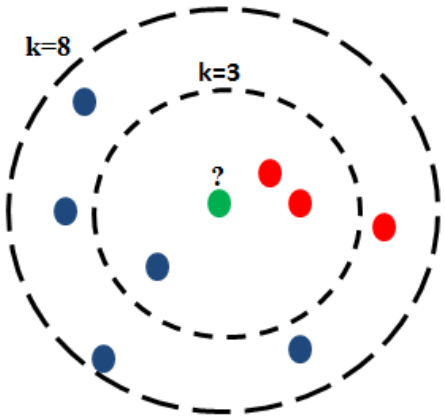


Figure 5.1 : k-nn algorithm for k=3.

5.2 Distance Metrics

To determine whether the person images belong to the same person or not, we compare the feature vectors obtained from these person images or image parts. There are several distance metrics to compare feature vectors. These functions generally take two feature vector and then produce a distance value. If the distance is higher then the threshold value, we can say that these two persons are not the same and if the distance is low, images can belong to the same person. At this point, distance and similarity are inversely proportional.

The distance metrics which have been used in this work, are listed below;

- Euclidean Distance, also known as the L2 distance, measures the lenght of straight-line between two points. The distance between two points in the plane with coordinates (X, Y) is given by,

$$d(X,Y) = \sqrt{\sum_{i=1}^{n}(x_i - y_j)^2} \tag{5.1}$$

- Cosine Distance, gives the angular cosine distance between vectors X and Y. It can be calculated as,

$$\cos(\theta) = \frac{XY}{|X||Y|} = \frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} x_i^2}\sqrt{\sum_{i=1}^{n} y_i^2}} \tag{5.2}$$

- Mahalanobis Distance measures the dissimilarity between two points of the same distribution with the covariance matrix C. It is calculated as,

$$d(X,Y) = \sqrt{(X-Y)^T C^{-1}(X-Y)} \tag{5.3}$$

6. EXPERIMENTAL RESULTS

In this chapter, the datasets that we have used in our experiments, pipeline of our model, experimental setup and performance of our proposed method is explained. Our proposed method is based on the cross dataset person re-identification setup. That means, we have used different datasets for training and test phases. In our experiments, we used CUHK03 [4] and Market-1501 [22] datasets as the training sets. To evaluate the performance of our method, we used VIPeR [9], PRID 2011 [23], and iLIDS-VID [24] datasets.

6.1 Datasets

As indicated previously in Section 3, the success rate of the deep learning based methods is highly related to the amount of training data that you have. Because of this reason, we used CUHK03 [4] and Market-1501 [22] datasets at the training stage. Currently, these are the largest datasets, which are used in person re-identification problem.

The CUHK03 [4] data set contains 14096 images of 1467 pedestrians, captured by five different surveillance cameras. Each person has two different security camera views. In this dataset there are two different types of person images: first one is manually labeled pedestrian bounding boxes and the second is automatically obtained bounding boxes by running a pedestrian detector. We used manually labeled pedestrian images for training purpose.

To our knowledge, Market-1501 [22] is currently the largest public data set of person re-identification. It contains 25259 images of 1501 pedestrians. These images are captured by six different surveillance cameras. In this data set, some of the images have been obtained by high resolution cameras (1280x1080) and the others by low resolution cameras (720x576). In some experiments, we combined CUHK03 [4] and Market-1501 [22] data sets to enlarge our training data. When we combined them, the softmax layers distribution was fine-tuned as 2968 different class labels.

The PRID 2011 [23] dataset consists of person images captured from two different static surveillance cameras (camera A and camera B). This dataset contains several challenges such as viewpoint and pose changes, differences in illumination, background and camera characteristics. Each camera has different number of person; camera A has 385 and camera B has 749. The first 200 persons appear in both camera views. So, there are 200 person image pairs in the dataset. The remaining persons in each camera view can be seen as gallery set. For evaluation on the test set, the following procedure is performed; we select camera A as a probe set and camera B as a gallery set. Each person, which is located in the probe set, is searched in the gallery set. We applied this procedure 10 times and then the results are reported as a CMC curve.

The other dataset, which we used in our experiments is iLIDS-VID dataset [24]. There are 300 different pedestrians in this dataset and all of them are captured from two different security cameras. The images are belong to airport arrival hall video security system. This system is a multi-camera surveillance system. There are 600 images in this dataset and each person has one pair of image sequences captured from two different cameras. Also, this dataset has some many difficulties such as clothing similarities among people, illumination and viewpoint changes, cluttered background and random occlusions. For evaluation purpose, we applied similar test procedure. We selected camera 1 as a probe set and camera 2 as a gallery set. Every person, which is come from probe set, searched in the gallery set. We applied this procedure 10 times and average results are reported.

The last dataset that we selected for our experiments is VIPeR [9] dataset. This is the most commonly used dataset in person re-identification problem. For this reason, we gave more attention to this dataset during our experiments. It contains 632 person image pairs taken from two different camera views. As in the other datasets, it contains illumination, pose, and viewpoint changes. For evaluation, this time, we followed a bit different procedure. This is also accepted evaluation procedure for VIPeR [9] dataset. 316 image pairs are randomly selected as the test set. Firstly, the image pairs are randomly assigned to probe and gallery sets. From the probe set, one image is randomly selected and then matched with all images from the gallery set. This operation is repeated by using every probe image. After all, this procedure is repeated

10 times and the average CMC curve is reported. Examples of all three datasets are shown in Figure 6.1.

Figure 6.1 : (a) Example image pairs from the PRID 2011 [23], (b) the iLIDS-VID (b) [24], and (c) the VIPeR [9].

6.2 Pipeline of the Proposed Method

In this work, we have proposed a deep learning based solution for person re-identification problem. This solution consists of basically two steps: traing and testing. Also we used some preprocessing methods to enhance our success rate. In this section, each step is explained in details. The entire system of our proposed method is shown in Figure 6.2.

6.2.1 Preprocessing

Before the selected CNNs have been fine-tuned, we applied some preprocessing techniques on the training and testing data. Firstly, to decrease the effect of illumination changes, we have performed histogram equalization method. It is a well-known method for contrast adjustment in image processing. Histogram equalization can be directly applied to gray scale images, since they are just single channel images. But now, RGB images consist of three channels. Histogram equalization cannot be applied on each image channel and then fused them back together. It is a non-linear process. The equalization procedure does not contain the color components; it contains intensity values of the image. For example, it is

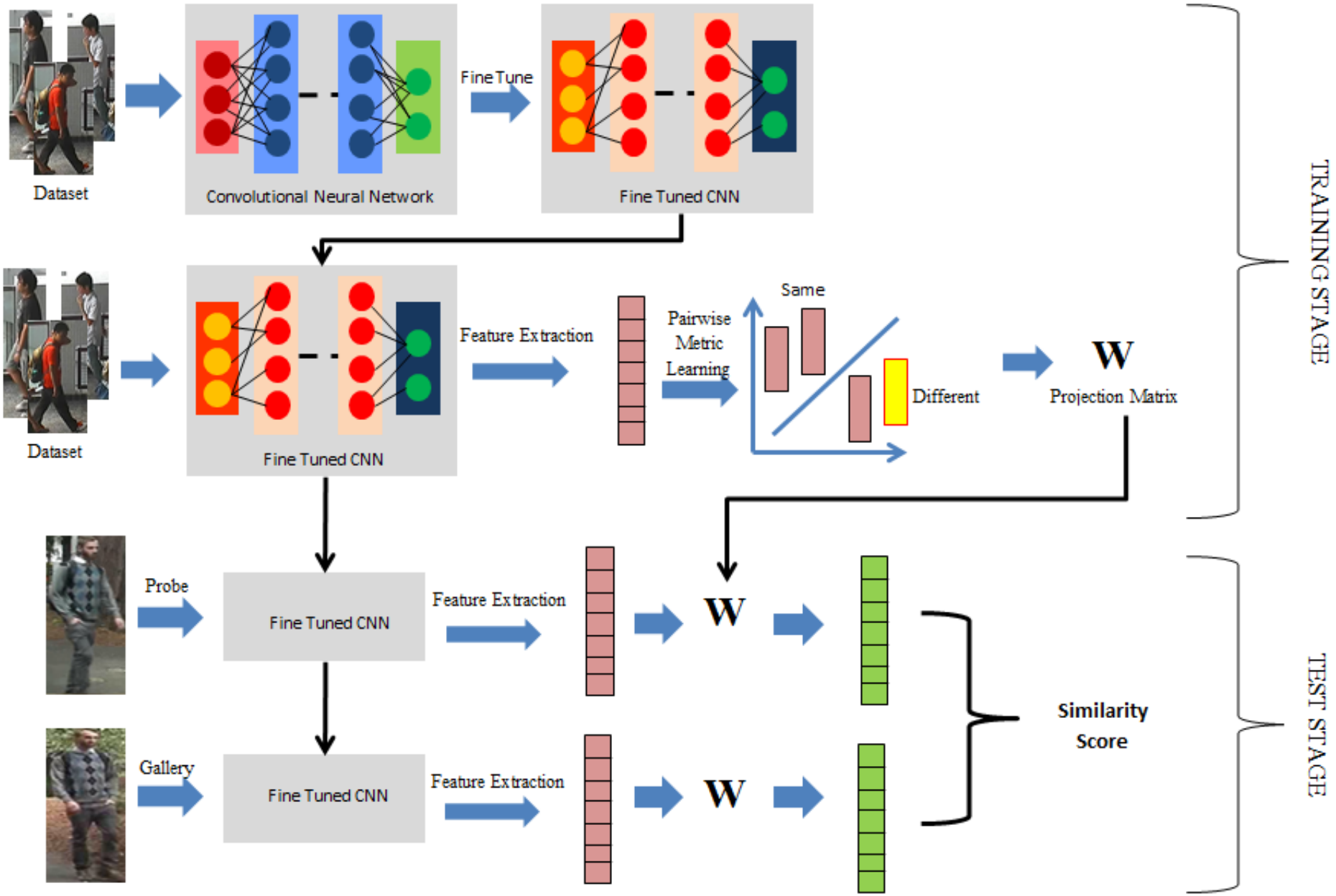


Figure 6.2 : Pipeline of our proposed method.

impossible to directly apply this operation to each channel of RGB image. We have to find a way that does not change the color balance of image. The operation has to handle only intensity values of images [41]. So firstly, we converted our images from RGB color space to HSV (Hue-Saturation-Value) color space. In HSV space, Hue represents the color type. Saturation represents the vibrancy of the color and Value represents the brightness of the color. The V band of the image is used for the histogram equalization. RGB images are converted to HSV color space by using Equation from (6.1) to (6.5) respectively.

$$R' = R/255, G' = G/255, B' = B/255 \tag{6.1}$$

$$C_{max} = max(R', G', B'), C_{min} = min(R', G', B'), \Delta = C_{max} - C_{min} \tag{6.2}$$

$$H = \begin{cases} 60^\circ \times (\frac{G' - B'}{\Delta} mod 6), & C_{max} = R' \\ 60^\circ \times (\frac{B' - R'}{\Delta} + 2), & C_{max} = G' \\ 60^\circ \times (\frac{R' - G'}{\Delta} + 4), & C_{max} = B' \end{cases} \tag{6.3}$$

$$S = \begin{cases} 0, & C_{max} = 0 \\ \frac{\Delta}{C_{max}}, & C_{max} \neq 0 \end{cases} \tag{6.4}$$

$$V = C_{max} \tag{6.5}$$

After color space has been changed, histogram equalization algorithm is applied. The general histogram equalization formula is;

$$h(v) = round(\frac{cdf(v) - cdf_{min}}{(MxN) - cdf_{min}} \times (L-1)) \tag{6.6}$$

where v is the pixel value, cdf is the cumulative distribution function, cdf_{min} is the minimum non-zero value of the cumulative distribution function, $M \times N$ gives the image's number of pixels and L is the number of levels used. This formula is applied to all pixel values in the image.

Next, we applied simple ellipse mask to get rid of unnecessary background objects. We put an ellipse at the center of the image and with respect to human body, we determined its height and weight values. And finally, images are divided into three non-coincident parts; head, body and legs. All preprocessing steps are summarized in Figure 6.3. Preprocessing steps have been applied to both training and test images.

Figure 6.3 : Prerocessing steps: **a)**original image **b)**histogram equalized image **c)**ellipse mask applied image **d)**division into three part.

6.2.2 Training

In this work, there are two different training stages. First one is fine-tuning procedure on CNNs and the second one is metric learning process. We trained several CNNs, which are mentioned previously in Section 3.2, by using CUHK03 [4] and Market-1501 [22] datasets. At this point, we use "training" phrase as fine-tuning the CNNs. Each deep neural network has been trained for each body part. Then, these CNNs have become ready to be used for feature extraction. During training process, CUHK03 [4] and Market-1501 [22] datasets are used separately and also together.

In order to extract features, again we use CUHK03 [4] and Market-1501 [22] datasets. These features are used in metric learning algorithms. Then, obtained features are labeled as positives and negatives whether they belong to same person or not. After that, using metric learning algorithms, projection matrices have been obtained. These matrices will transfer our test features to more discriminative feature space during performance evaluation.

6.2.3 Test

To evaluate our proposed method, we used VIPeR [9] dataset. According to test perocedure, which was described in Section 6.1, person images are passed through the fine-tuned neural networks. Their extracted features are transferred to new space using learned projection matrix. Then, their similarities are calculated in that space. The results are shown in average cumulative matching curves.

Aside from metric learning test procedure, we also reported direct similarities of extracted features to evaluate discriminative power of fine tuned networks. For this

pupose, we used VIPeR [9], PRID 2011 [23], and iLIDS-VID [24] datasets. After their features are extracted from CNNs, we directly compare their cosine similarities based on probe-gallery procedure. We also reported these results by using Cummulative Matching Curves.

6.3 Results

In this section, the results of the experiments are shown in detail. Metric learning and direct comparison results are reported separately. Several experiments are conducted on the mentioned datasets. VIPeR [9] has been used to show metric learning performance, PRID 2011 [23], iLIDS-VID [24] and again VIPeR [9] have been used to show direct similarity comparison. Moreover, different CNN layers' performance is reported in detail.

6.3.1 Metric learning results

In this work, we used Joint similarity and Diagonal metric learning methods. After the CNNs have been fine-tuned, extracted features from training datasets were used to learn projection matrices. As described previously in Section 4.2, firstly, positive and negative pairs were made ready. We prepared approximately 600k positive and negative pairs for each body parts. Then, using metric learning algorithms, projection matrices (W) have been learned. To evaluate their success rate, VIPeR [9] dataset have been used as described in Section 6.1. In order to implement cross dataset procedure, we have learned suitable metrics from CUHK03 [4] and Market-1501 [22] datasets, but then we tested on VIPeR [9] dataset. In this work, Cumulative Matching Characteristic (CMC) curves are used to report the average results. This curve shows the accuracy of finding the true match within the first r ranks. The obtained results are shown in Figure 6.4 and Figure 6.5, respectively. We also learned the metrics from VIPeR [9] dataset and then tested on VIPeR [9] again. For this procedure, we randomly divided the dataset into two part and half of them used for trainig purpose. While were doing this, we changed the learned metric's dimension to see the it's effect on recoginition rate. The detailed results can be found in Appendix A.2.

In Figure 6.4, the best achieved result is %18 at rank-1 position. It belongs to "middle" part of person images. This figure shows that the most discriminative part of person

images is middle part and on the contrary, legs are the least discriminative part. In Figure 6.5, we can see the diagonal metric results on VIPeR [9] dataset. The best achieved result is %14 at rank-1 position and it belongs to "middle" part of person images again. Because, leg parts of person images very similar to each others.

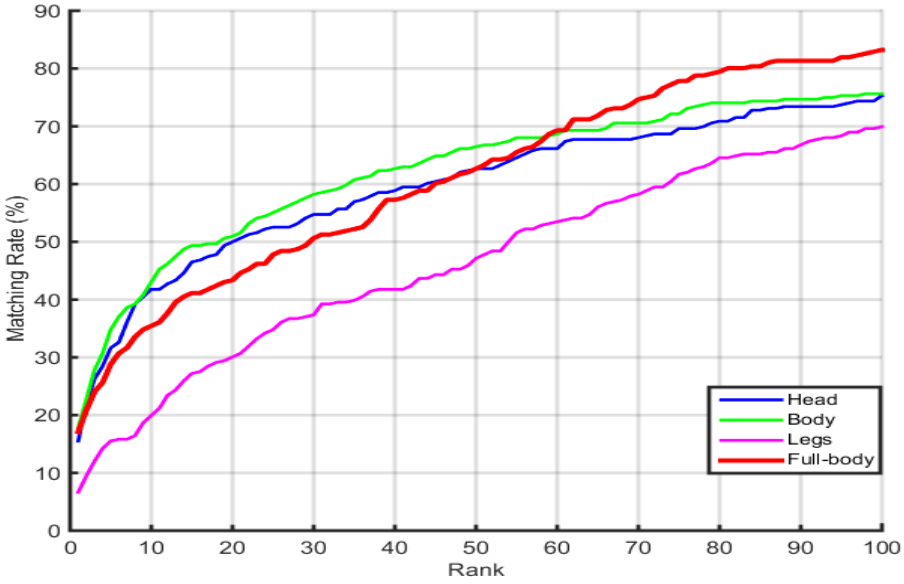


Figure 6.4 : Joint metric results on VIPeR [9].

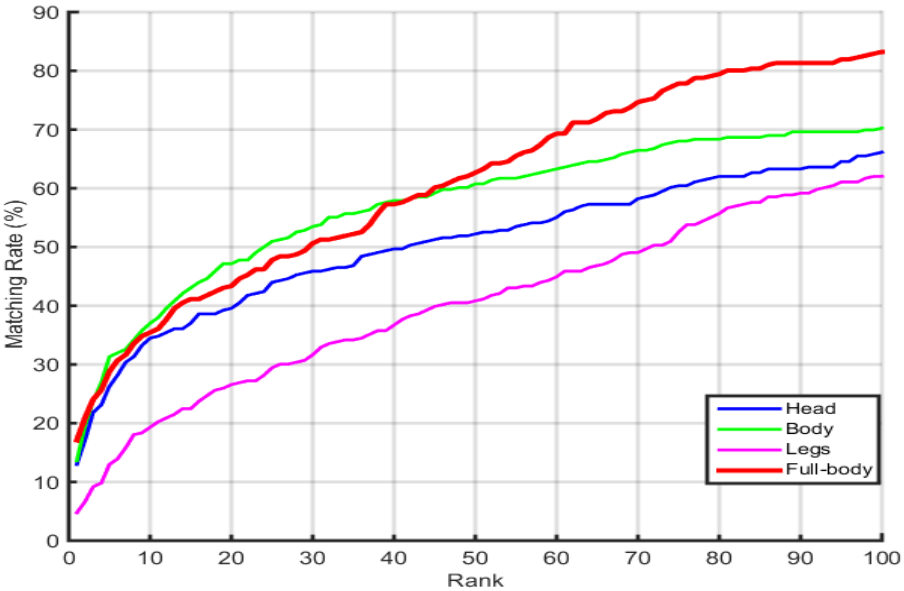


Figure 6.5 : Diagonal metric results on VIPeR [9].

6.3.2 Distance metric results

To evaluate the discriminative power of fine-tuned networks, besides the metric learning methods, we also measured the direct similarity between probe and gallery images. The extracted features have been used for this purpose. In our experiments we used different features, which are extracted from different layers of CNNs. To measure similarity between extracted features, cosine similarity metric is used. VIPeR [9], PRID 2011 [23], and iLIDS-VID [24] datasets are used as test sets for this section. We measured the distance between probe and gallery images ten times and then, average results have been reported.

6.3.2.1 Results on VIPeR dataset

Detailed results of VIPeR [9] dataset are shown in Table 6.1. To obtain these results, we fused the head, body and leg features, which are come from same person and then compare the others. Different CNNs and their different layers have been used to extract features. According to cosine similarity between probe and gallery set, their matching score is reported on different Rank values.

Table 6.1 : Fused results with different Rank (r) values on VIPeR [9].

DataSet	CNN	Layer	r=1(%)	30(%)	50(%)	100(%)
None	AlexNet [19]	FC7	15.2	51.3	61.6	76.1
None	AlexNet [19]	FC6	14.1	50.2	63.7	81.4
CUHK03 [4]	AlexNet [19]	FC7	26.3	67.4	75.7	84.2
CUHK03 [4]	AlexNet [19]	FC6	27.7	64.7	75.3	84.3
Market-1501 [22]	AlexNet [19]	FC7	27.1	68.2	74.3	82.3
Market-1501 [22]	AlexNet [19]	FC6	27.5	63.0	73.5	82.2
CUHK03 + Market-1501	AlexNet [19]	FC7	**32.4**	**68.6**	**77.2**	**87.1**
CUHK03 + Market-1501	AlexNet [19]	FC6	29.2	64.3	74.4	85.1
None	VGG-16 [20]	FC7	23.1	58.0	64.5	78.8
None	VGG-16 [20]	FC6	21.3	54.0	56.5	77.5
CUHK03 [4]	VGG-16 [20]	FC7	29.1	68.0	74.5	82.8
CUHK03 [4]	VGG-16 [20]	FC6	27.3	68.0	76.5	82.5
Market-1501 [22]	VGG-16 [20]	FC7	26.8	60.1	69.3	80.0
Market-1501 [22]	VGG-16 [20]	FC6	19.9	56.6	63.2	79.7
CUHK03 + Market-1501	VGG-16 [20]	FC7	30.4	65.6	71.2	84.0
CUHK03 + Market-1501	VGG-16 [20]	FC6	30.2	61.3	70.4	81.1
None	GoogLeNet [21]	pool5_7x7_s1	18.0	52.2	60.2	71.8
CUHK03 [4]	GoogLeNet [21]	pool5_7x7_s1	24.0	62.2	70.2	77.8
Market-1501 [22]	GoogLeNet [21]	pool5_7x7_s1	25.3	64.5	71.5	82.2

As can be seen in Table 6.1, we achieved the best result by using AlexNet [19] deep neural network, which has been fine-tuned with CUHK03 [4] and Market-1501 [22] datasets. The matching score is %32.4 at the Rank-1 and it grows cumulatively. The recognition rates are compared with other methods in Table 6.2.

Table 6.2 : Comparison with the other methods on VIPeR [9].

Methods	Training sets	Rank1%	Rank10%	Rank20%	Rank30%
DTRSVM [42]	i-LIDS	8.26	31.38	44.83	53.88
DTRSVM [42]	PRID	10.90	28.20	37.69	44.87
DML [16]	CUHK Campus	16.17	45.82	57.56	64.24
CDPR [43]	CUHK02	16.90	45.89	58.58	65.32
CDPR [43]	CASPR	17.22	48.01	57.56	64.15
CDPR [43]	Shinpuhkan2014	18.64	48.54	60.63	68.48
Ours	CUHK03 + Market-1501	**32.4**	**51.58**	**60.12**	**68.60**

From Table 6.2, it can be observed that, our results outperform the existing cross dataset methods significantly. We have achieved 32% matching rate at Rank-1 result. These results are achieved by using the fused extracted features. The head, body, and legs results can be seen individually in Figure 6.6.

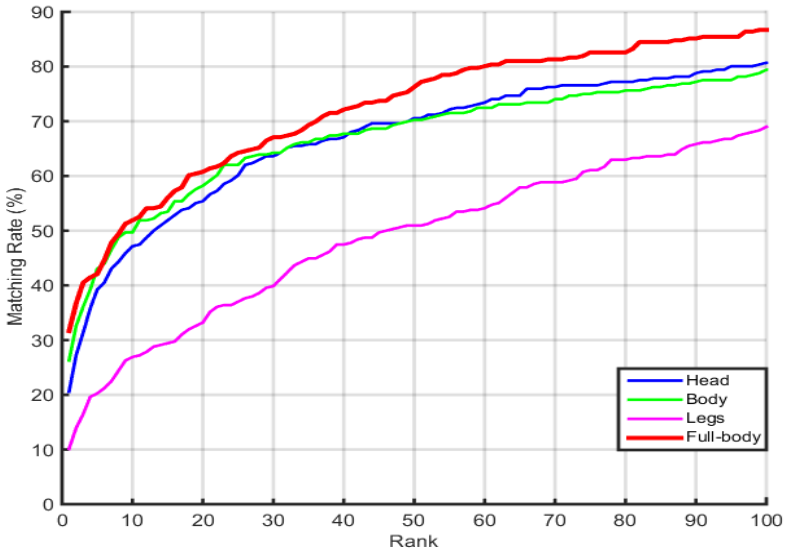


Figure 6.6 : The CMC curve on the VIPeR [9] dataset, the training sets are CUHK03 [4] and Market-1501 [22].

From the results, we can see that the cross dataset evaluation accuracies of person re-identification are currently very low. But this topic is very important, especially in practical applications. For our method, we learned essential parameters from different datasets and we use this knowledge to match person images, which are obtained from

different datasets. Nevertheless, our fused results on VIPeR [9] is satisfying. As shown in Figure 6.6, we can say that the legs are the least important part to distinguish persons from each other.

6.3.2.2 Results on PRID 2011 dataset

The next experiment is performed on the PRID 2011 [23] dataset. The evaluation protocol was performed as mentioned previously in Section 6.1. We used cosine similarity to measure distance between probe and gallery images. These features are extracted from AlexNet [19] neural network. Full body matching results are shown in Table 6.3. Also, the different layer's performances can be seen from the table.

Table 6.3 : Fused results with different Rank (r) values on PRID 2011 [23].

DataSet	CNN	Layer	r=1(%)	30(%)	50(%)	100(%)
None	AlexNet [19]	FC7	2.2	15.3	21.2	26.5
None	AlexNet [19]	FC6	**2.7**	**18.8**	**21.6**	**30.4**
CUHK03 [4]	AlexNet [19]	FC7	2.5	5.5	9.4	10.5
CUHK03 [4]	AlexNet [19]	FC6	4	7.6	6.5	10.0
Market-1501 [22]	AlexNet [19]	FC7	1.5	6.5	7.5	12.0
Market-1501 [22]	AlexNet [19]	FC6	2.8	7.3	9.6	11.7

From the Table 6.3, we can say that, pure AlexNet [19] network gives better results on PRID [23] dataset. The results are not well at low Rank values, but it gives better performance at higher Ranks. The best result is 30% at Rank-100 value. Figure 6.7 shows the rank curves of the three parts and their fusion. From the results we can see that the PRID is a very difficult dataset. The main reason is that chromatic aberration of the images in camera B folder [43].

6.3.2.3 Results on iLIDS-VID dataset

The last test dataset is iLIDS-VID [24]. This set involves 300 different persons and each of them has two images captured by two different cameras. Each person's features are extracted from fine-tuned AlexNet [19] neural network. Their similarities are calculated by using cosine distance metric. Different layers and different training sets have been used in this setup. The performances are summarized for different Rank values in Table 6.4.

For this experiment we used AlexNet [19] neural network. Extracted features which are acquired from FC7 and FC6 layers, compared to each other by using cosine metric.

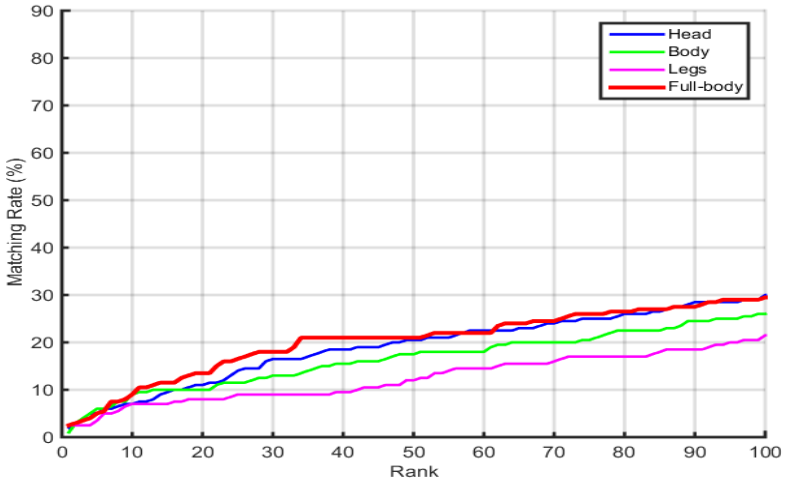


Figure 6.7 : The CMC curve on the PRID [23] dataset.

Table 6.4 : Fused results with different Rank (r) values on iLIDS-VID [24].

DataSet	CNN	Layer	r=1(%)	30(%)	50(%)	100(%)
None	AlexNet [19]	FC7	3.5	24.7	31.3	42.7
None	AlexNet [19]	FC6	4.8	25.8	32.7	45.7
CUHK03 [4]	AlexNet [19]	FC7	3.4	27.6	35.7	44.9
CUHK03 [4]	AlexNet [19]	FC6	4.3	27.8	33.6	47.8
Market-1501 [22]	AlexNet [19]	FC7	4.5	23.6	30.2	43.4
Market-1501 [22]	AlexNet [19]	FC6	5.4	25.5	32.2	46.9

The performance of the head part almost equals to fusion. As usual, the performance of the leg part is poor. Especially in this dataset, many people have suitcases and this is a handicap for the system. We can say that, Rank-50 results quite well on this dataset. Also the performance of each body part can be seen on Figure 6.8.

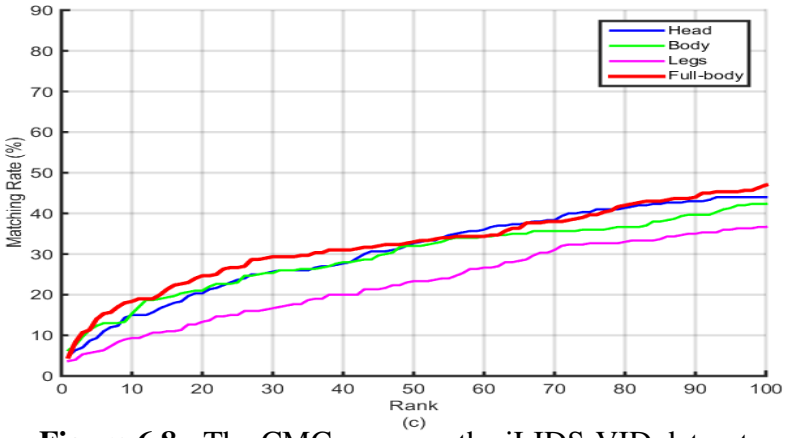


Figure 6.8 : The CMC curve on the iLIDS-VID dataset.

7. CONCLUSIONS AND FUTURE RECOMMENDATIONS

In this thesis, we proposed a deep learning based solution for person re-identification problem. Person re-identification can be defined as the process of matching individuals' images, which are obtained from different camera views. Due to the use of different cameras, the problem contains different lighting, pose and background conditions. These are the challenges that makes this problem hard. Also, person re-identification is one of the important issues in today's video surveillance systems.

With the development of GPU based technologies, recently, deep learning methods have achieved significant success in different areas. Person re-identification is one of them. But, the success rate of deep learning based methods is highly related to the amount of training data that one has. To cope with this, we used the largest datasets which are produced for the person re-identification problem. We also applied a transfer learning technique to improve compatibility capacity of these neural networks.

In this thesis, our main contribution is to show that several domain specific convolutional neural networks can be used also in person re-identification problem. We used the largest datasets in our experiments but these are not sufficient to train a whole convolutional neural network by alone. Due to this reason, for us, transfer learning is a good choice that should be considered. Therefore, we took pre-trained neural networks and adapt them into our problem. These networks have been originally developed for image recogniton problem. So, we fine-tuned these networks with the person re-identification datasets. While we were doing this, we kept the early layers' weights as much as possible and we changed the last layers more. Because, in convolutional neural networks, early layers correspond to more generic features. On the contrary, the last layers represent problem specific features. After fine-tuning operation, we analyzed the accuracy-loss graphics and we realized that these networks can be used for our problem successfully.

After fine-tuning operation, the convolutional neural networks have become ready to extract good features from person images. We used two different metric learning

algorithms with these features. The purpose of metric learning algorithms is to move extracted features to a more discriminative space. This is done via projection matrices, which are obtained from training data. Respectively, we extracted our features, learned projection matrices, and then used these matrices to evaluate success rate of metric learning methods. With metric learning algorithms, we achieved good results at different rank values. Several datasets have been used in test phase. Besides, metric learning methods, we also compared extracted features directly. We used cosine similarity metric to measure distances between probe and gallery images. We achieved satisfactory results by using cosine similarity, especially on VIPeR [9] dataset. The results are reported as average CMC curves.

Despite to the many advances in this field, person re-identification is still an open problem. There is no generic solution to cover all kind of datasets and conditions. In this thesis, we tried to cope with cross dataset person re-identification problem. This means, we used different datasets at training and test stages. This procedure complicates the problem even more.

For the future work, different metric learning algorithms can be used to increase discriminative power of extracted features. Finding good feature representation is important in image matching problems. Additionally, with the increase of person re-identification datasets, convolutional neural networks may become more successful to learn human appearance. Currently, there are not large enough datasets to train a deep neural network from scratch. It should be noted that, the cross dataset person re-identification problem is more important than the same dataset re-identification problem. Because, in real world applications it is hard to collect person images from the same domain. Finally, in terms of video surveillance systems, person re-identification is very important subject. Solving this problem in real time conditions will provide a great contribution to ensure public safety.

REFERENCES

[1] **Hampapur, A.**, **Brown, L.**, **Connell, J.**, **Pankanti, S.**, **Senior, A. and Tian, Y.** (2003). Smart surveillance: applications, technologies and implications.

[2] **Vezzani, R.**, **Baltieri, D. and Cucchiara, R.** (2013). People reidentification in surveillance and forensics: A survey, *ACM Computing Surveys (CSUR)*, *46*(2), 29.

[3] **Saghafi, M.A.**, **Hussain, A.**, **Zaman, H.B. and Saad, M.H.M.** (2014). Review of person re-identification techniques, *IET Computer Vision*, *8*(6), 455–474.

[4] **Li, W.**, **Zhao, R.**, **Xiao, T. and Wang, X.** (2014). DeepReid: Deep Filter Pairing Neural Network for Person Re-identification, *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Columbus, USA.

[5] **Gong, S.**, **Cristani, M.**, **Yan, S. and Loy, C.C.** (2014). *Person re-identification*, volume 1, Springer.

[6] **Hamdoun, O.**, **Moutarde, F.**, **Stanciulescu, B. and Steux, B.** (2008). Person re-identification in multi-camera system by signature based on interest point descriptors collected on short video sequences, *2nd ACM/IEEE International Conference on Distributed Smart Cameras (ICDSC-08)*, Stanford, Palo Alto, United States, pp. –, `https://hal.inria.fr/inria-00332032`.

[7] **Bąk, S.**, **Corvee, E.**, **Bremond, F. and Thonnat, M.** (2012). Boosted human re-identification using riemannian manifolds, *Image and Vision Computing*, *30*(6), 443–452.

[8] **Farenzena, M.**, **Bazzani, L.**, **Perina, A.**, **Murino, V. and Cristani, M.** (2010). Person re-identification by symmetry-driven accumulation of local features, *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, IEEE, pp.2360–2367.

[9] **Gray, D.**, **Brennan, S. and Tao, H.** (2007). Evaluating Appearance Models for Recognition, Reacquisition, and Tracking, *Proc. IEEE International Workshop on Performance Evaluation for Tracking and Surveillance (PETS)*.

[10] **D'angelo, A. and Dugelay, J.L.** (2011). People re-identification in camera networks based on probabilistic color histograms, *Proc. SPIE*, volume7882.

[11] **Li, Y.**, **Wu, Z. and Radke, R.J.** (2015). Multi-shot re-identification with random-projection-based random forests, *2015 IEEE Winter Conference on Applications of Computer Vision*, IEEE, pp.373–380.

[12] **Weinberger, K.Q. and Saul, L.K.** (2009). Distance metric learning for large margin nearest neighbor classification, *Journal of Machine Learning Research*, *10*(Feb), 207–244.

[13] **Köstinger, M.**, **Hirzer, M.**, **Wohlhart, P.**, **Roth, P.M. and Bischof, H.** (2012). Large scale metric learning from equivalence constraints, *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, IEEE, pp.2288–2295.

[14] **Li, Z.**, **Chang, S.**, **Liang, F.**, **Huang, T.S.**, **Cao, L. and Smith, J.R.** (2013). Learning locally-adaptive decision functions for person verification, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp.3610–3617.

[15] **Davis, J.V.**, **Kulis, B.**, **Jain, P.**, **Sra, S. and Dhillon, I.S.** (2007). Information-theoretic metric learning, *Proceedings of the 24th international conference on Machine learning*, ACM, pp.209–216.

[16] **Yi, D.**, **Lei, Z. and Li, S.Z.** (2014). Deep metric learning for practical person re-identification, *arXiv preprint arXiv:1407.4979*.

[17] **Ahmed, E.**, **Jones, M. and Marks, T.K.** (2015). An improved deep learning architecture for person re-identification, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp.3908–3916.

[18] **Url-1**, `https://en.wikipedia.org/wiki/Inductive_transfer`, date retrieved: 29.04.2016.

[19] **Krizhevsky, A.**, **Sutskever, I. and Hinton, G.E.** (2012). Imagenet classification with deep convolutional neural networks, *Advances in neural information processing systems*, pp.1097–1105.

[20] **Simonyan, K. and Zisserman, A.** (2014). Very deep convolutional networks for large-scale image recognition, *arXiv preprint arXiv:1409.1556*.

[21] **Szegedy, C.**, **Liu, W.**, **Jia, Y.**, **Sermanet, P.**, **Reed, S.**, **Anguelov, D.**, **Erhan, D.**, **Vanhoucke, V. and Rabinovich, A.** (2015). Going deeper with convolutions, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp.1–9.

[22] **Zheng, L.**, **Shen, L.**, **Tian, L.**, **Wang, S.**, **Bu, J. and Tian, Q.** (2015). Person re-identification meets image search, *arXiv preprint arXiv:1502.02171*.

[23] **Hirzer, M.**, **Beleznai, C.**, **Roth, P.M. and Bischof, H.** (2011). Person Re-Identification by Descriptive and Discriminative Classification, *Proc. Scandinavian Conference on Image Analysis (SCIA)*, www.springerlink.com.

[24] **Wang, T.**, **Gong, S.**, **Zhu, X. and Wang, S.**, (2014). Person re-identification by video ranking, Computer Vision–ECCV 2014, Springer, pp.688–703.

[25] **Url-2**, `https://en.wikipedia.org/wiki/Deep_learning`, date retrieved: 29.04.2016.

[26] **LeCun, Y.**, **Bottou, L.**, **Bengio, Y. and Haffner, P.** (2001). Gradient-Based Learning Applied to Document Recognition, *Intelligent Signal Processing*, IEEE Press, pp.306–351.

[27] **van Doorn, J.** (2014). Analysis of Deep Convolutional Neural Network Architectures.

[28] **Url-3**, `https://imiloainf.files.wordpress.com/2013/11/activation_funcs1.png`, date retrieved: 29.04.2016.

[29] **Url-4**, `https://upload.wikimedia.org/wikipedia/commons/e/e9/Max_pooling.png`, date retrieved: 29.04.2016.

[30] **Bouchain, D.** (2006). Character recognition using convolutional neural networks, *Institute for Neural Information Processing*, *2007*.

[31] **Url-8**, `http://www.cse.unsw.edu.au/~cs9417ml/MLP2/`, date retrieved: 14.08.2016.

[32] **Lu, W.** (2000). *Neural network model for distortional buckling behaviour of cold-formed steel compression members*, Helsinki University of Technology.

[33] **Deng, J.**, **Dong, W.**, **Socher, R.**, **Li, L.J.**, **Li, K. and Fei-Fei, L.** (2009). ImageNet: A large-scale hierarchical image database, *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pp.248–255.

[34] **Torrey, L. and Shavlik, J.** (2009). Transfer learning, *Handbook of Research on Machine Learning Applications and Trends: Algorithms, Methods, and Techniques*, *1*, 242.

[35] **Pan, S.J. and Yang, Q.** (2010). A Survey on Transfer Learning, *IEEE Transactions on Knowledge and Data Engineering*, *22*(10), 1345–1359.

[36] **Url-5**, `http://cs231n.github.io/transfer-learning/#tf`, date retrieved: 29.04.2016.

[37] **Url-6**, `https://en.wikipedia.org/wiki/Feature_extraction`, date retrieved: 29.04.2016.

[38] **Url-9**, `http://www.mathworks.com/discovery/feature-extraction.html?requestedDomain=www.mathworks.com`, date retrieved: 14.08.2016.

[39] **Url-7**, `https://en.wikipedia.org/wiki/Similarity_learning`, date retrieved: 29.04.2016.

[40] **Ghaleb, E.**, **Tapaswi, M.**, **Al-Halah, Z.**, **Ekenel, H.K. and Stiefelhagen, R.** (2015). Accio: A Data Set for Face Track Retrieval in Movies Across Age, *Proceedings of the 5th ACM on International Conference on Multimedia Retrieval*, ACM, pp.455–458.

[41] **Url-10**, https://prateekvjoshi.com/2013/11/22/histogram-equalization-of-rgb-images/, date retrieved: 14.08.2016.

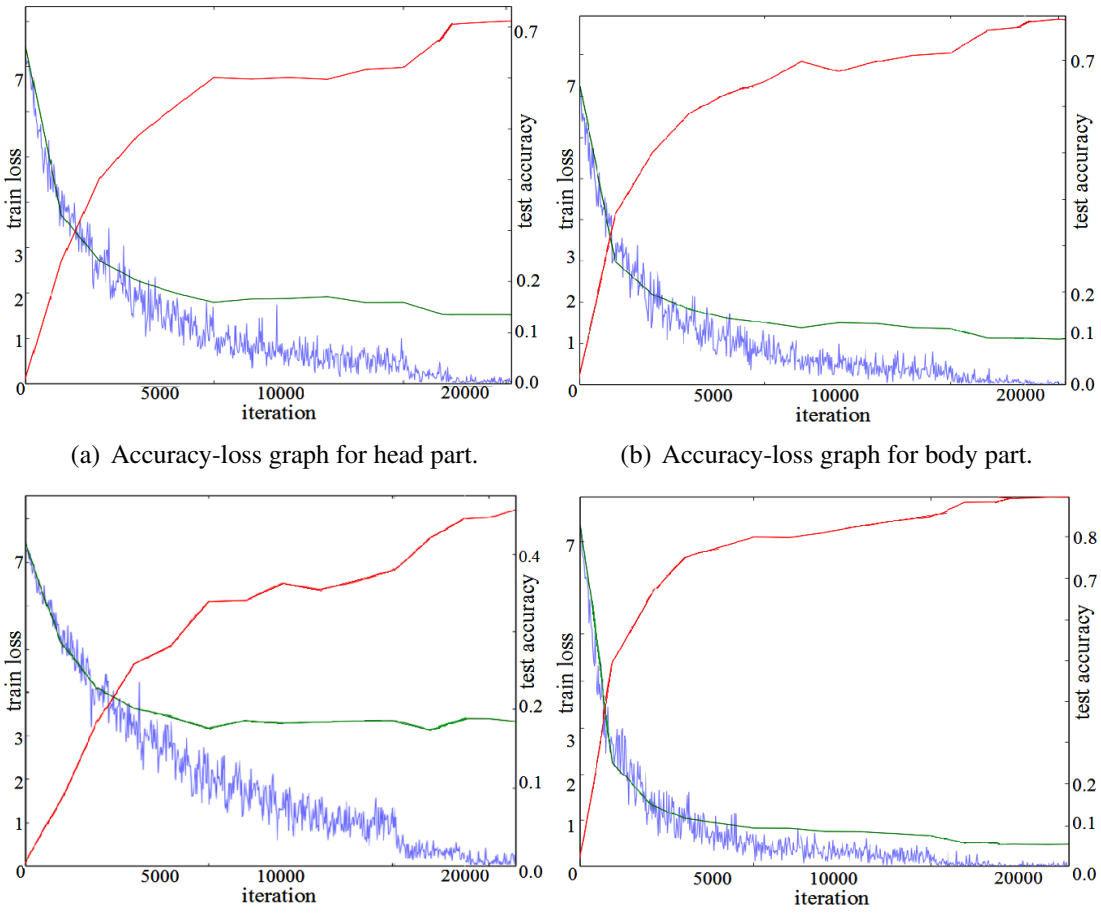
[42] **Ma, A.J.**, **Yuen, P.C. and Li, J.** (2013). Domain Transfer Support Vector Ranking for Person Re-identification without Target Camera Label Information, *2013 IEEE International Conference on Computer Vision*, pp.3567–3574.

[43] **Hu, Y.**, **Yi, D.**, **Liao, S.**, **Lei, Z. and Li, S.Z.** (2014). Cross dataset person re-identification, *Computer Vision-ACCV 2014 Workshops*, Springer, pp.650–664.

APPENDICES

APPENDIX A.1 : Accuracy-Loss Graphics
APPENDIX A.2 : Metric Learning Results

APPENDIX A.1



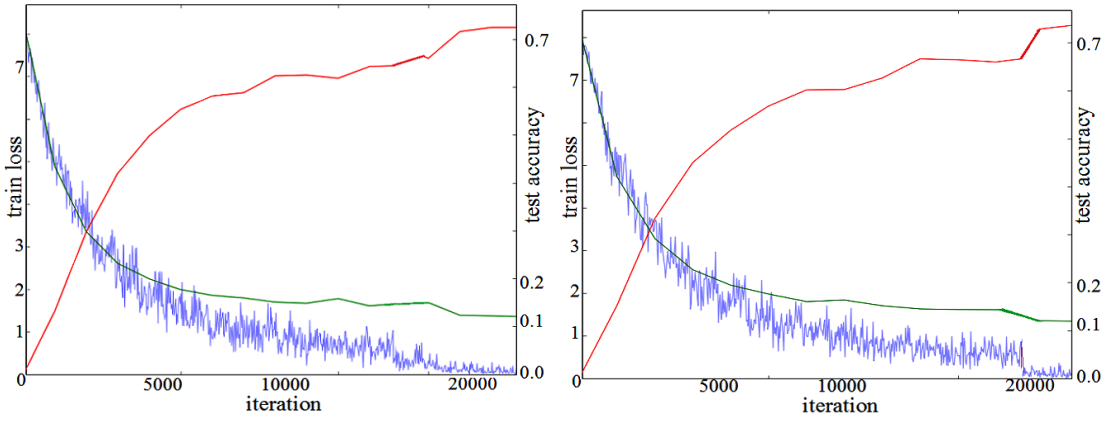
(a) Accuracy-loss graph for head part.

(b) Accuracy-loss graph for body part.

(c) Accuracy-loss graph for legs part.

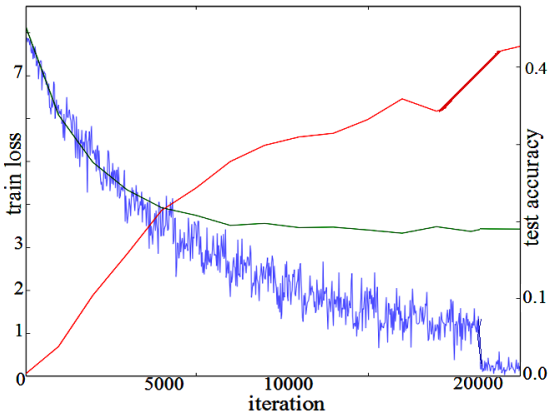
(d) Accuracy-loss graph for full-body.

Figure A.1 : AlexNet [19] fine-tuning results for Market-1501 [22] dataset.

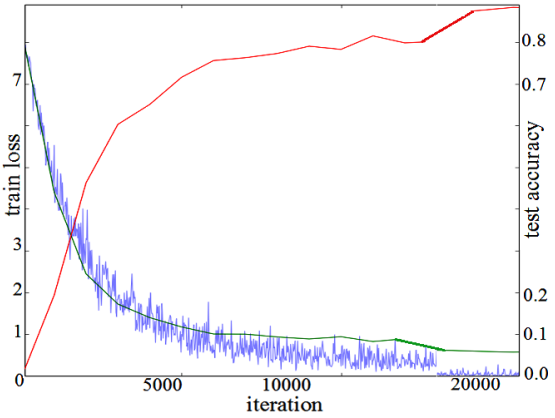


(a) Accuracy-loss graph for head part.

(b) Accuracy-loss graph for body part.

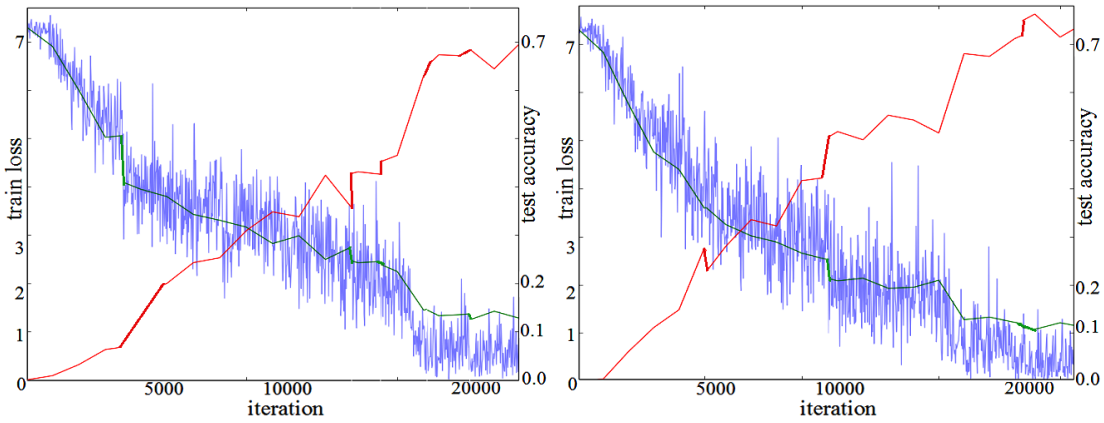


(c) Accuracy-loss graph for legs part.



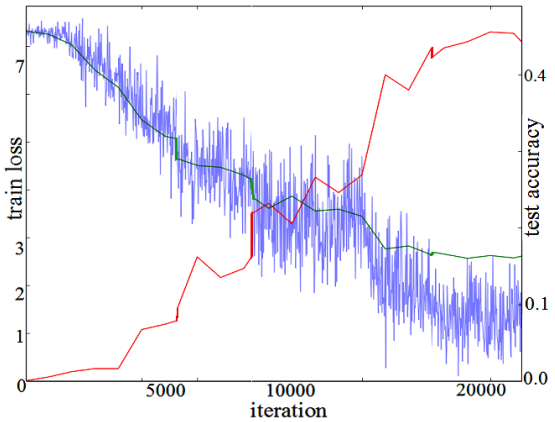
(d) Accuracy-loss graph for full-body.

Figure A.2 : AlexNet [19] fine-tuning results for the union of Market-1501 [22] and CUHK03 [4] datasets.

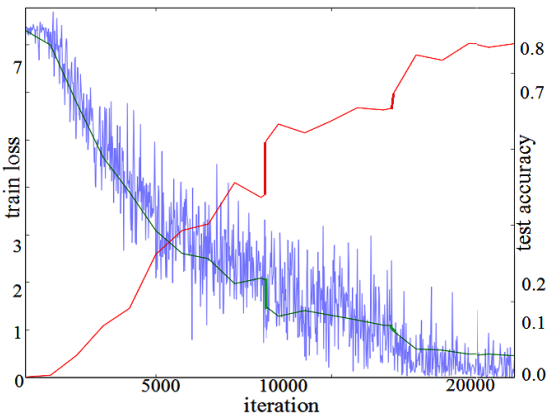


(a) Accuracy-loss graph for head part.

(b) Accuracy-loss graph for body part.

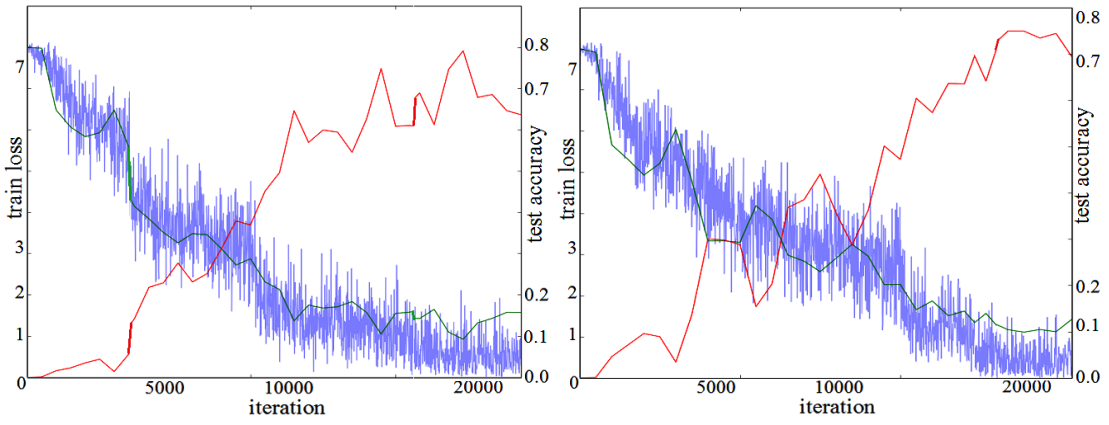


(c) Accuracy-loss graph for legs part.



(d) Accuracy-loss graph for full-body.

Figure A.3 : VGG-16 [20] fine-tuning results for Market-1501 [22] dataset.



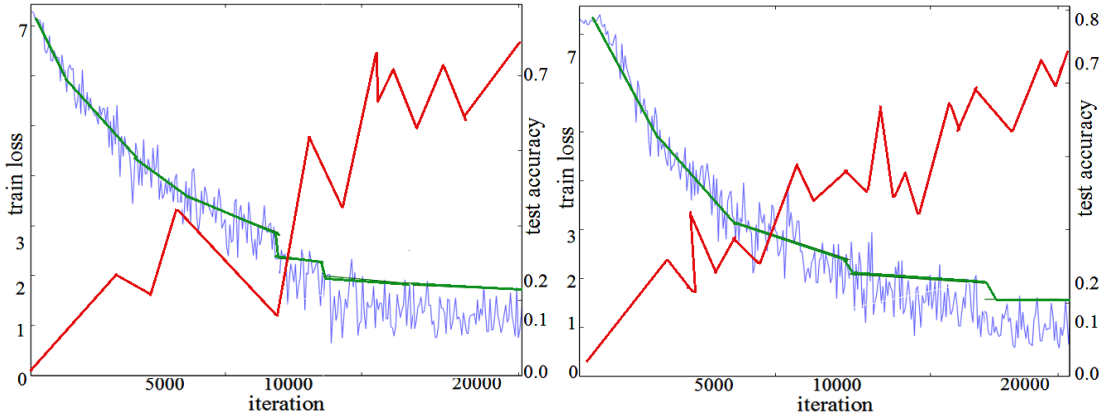
(a) Accuracy-loss graph for head part.

(b) Accuracy-loss graph for body part.

(c) Accuracy-loss graph for legs part.

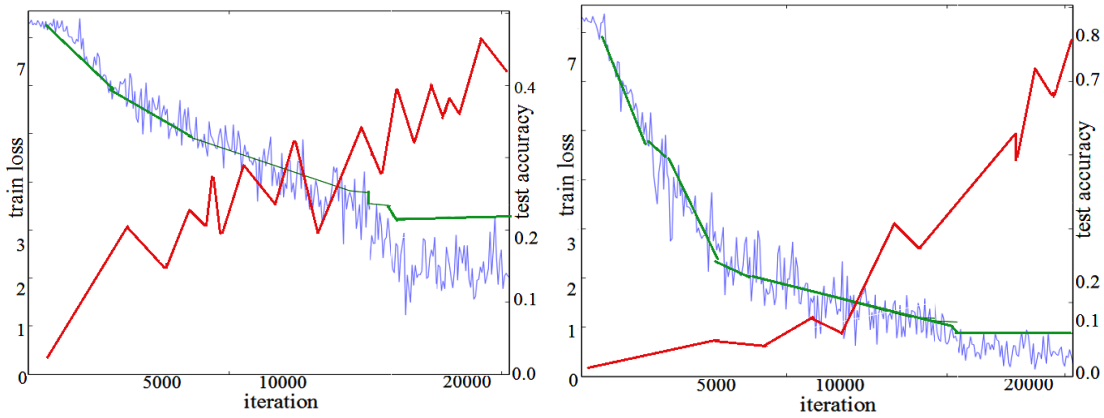
(d) Accuracy-loss graph for full-body.

Figure A.4 : VGG-16 [20] fine-tuning results for the union of Market-1501 [22] and CUHK03 [4] datasets.



(a) Accuracy-loss graph for head part.

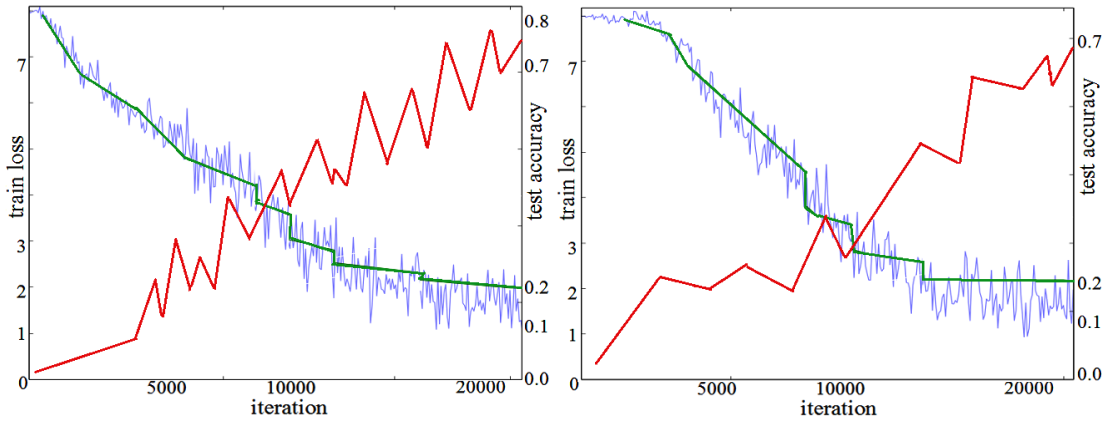
(b) Accuracy-loss graph for body part.



(c) Accuracy-loss graph for legs part.

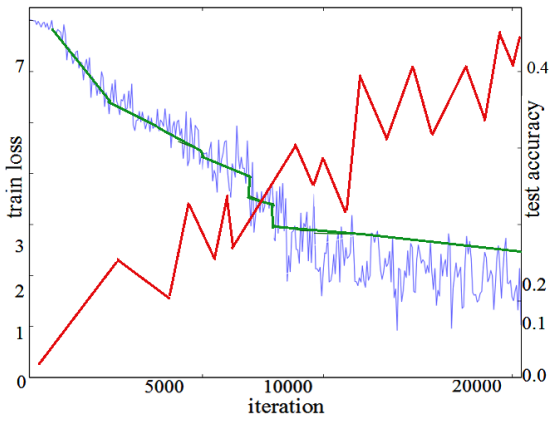
(d) Accuracy-loss graph for full-body.

Figure A.5 : GoogLeNet [21] fine-tuning results for Market-1501 [22] dataset.

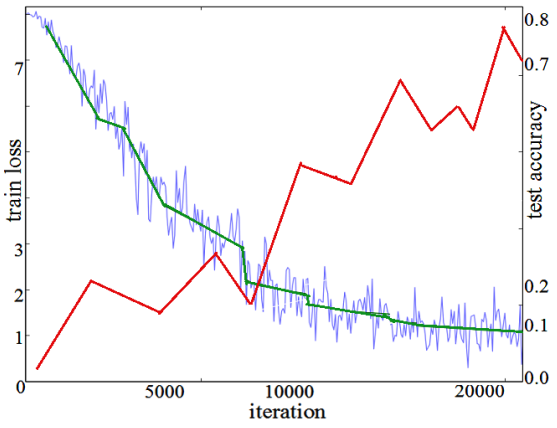


(a) Accuracy-loss graph for head part.

(b) Accuracy-loss graph for body part.



(c) Accuracy-loss graph for legs part.



(d) Accuracy-loss graph for full-body.

Figure A.6 : GoogLeNet [21] fine-tuning results for the union of Market-1501 [22] and CUHK03 [4] datasets.

APPENDIX A.2

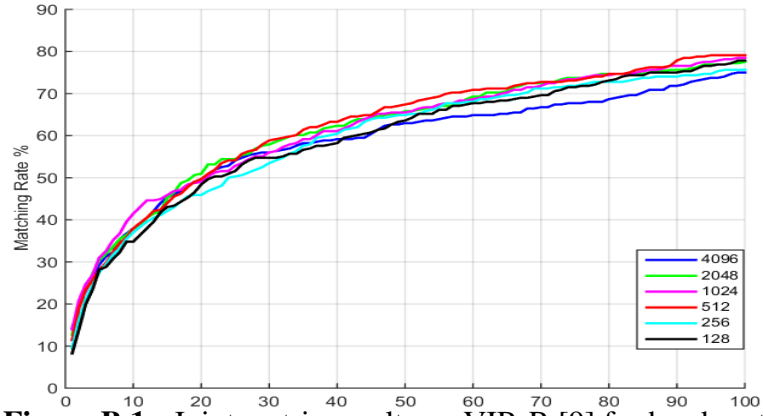


Figure B.1 : Joint metric results on VIPeR [9] for head part.

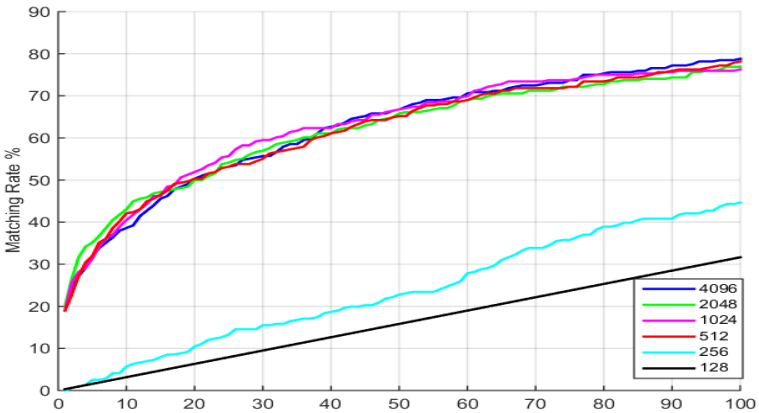


Figure B.2 : Joint metric results on VIPeR [9] for body part.

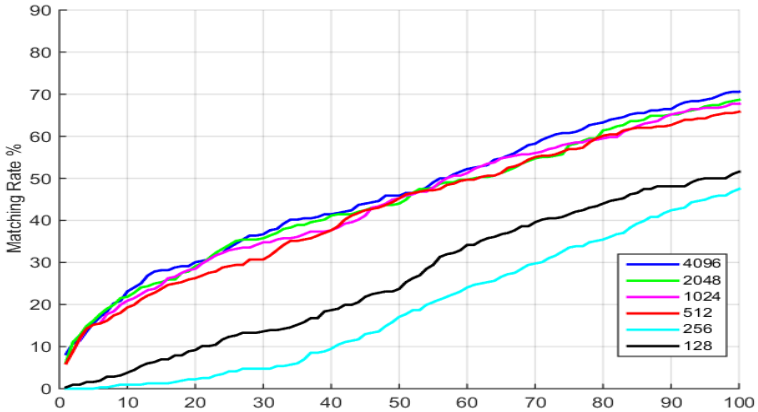


Figure B.3 : Joint metric results on VIPeR [9] for leg part.

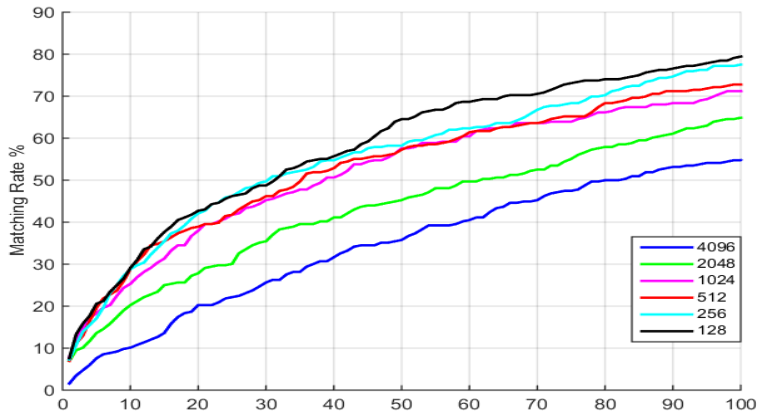


Figure B.4 : Diagonal metric results on VIPeR [9] for head part.

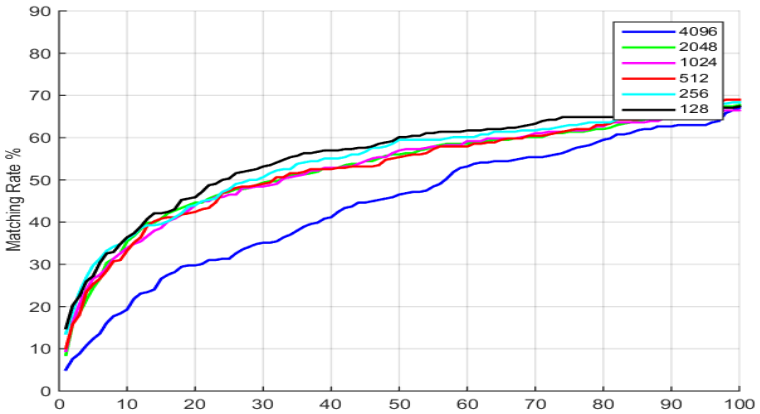


Figure B.5 : Diagonal metric results on VIPeR [9] for body part.

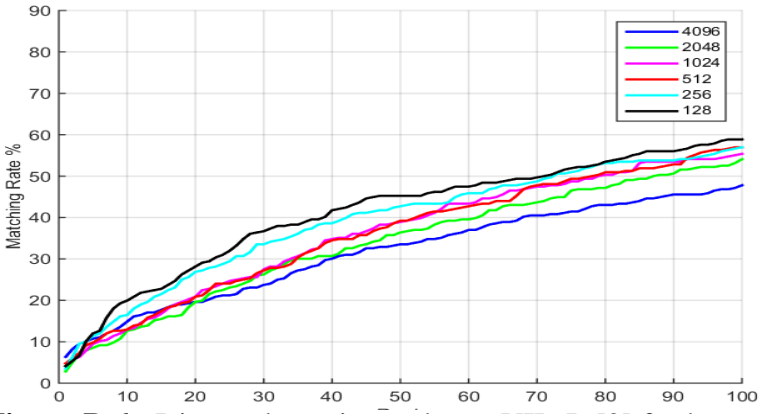


Figure B.6 : Diagonal metric results on VIPeR [9] for leg part.

CURRICULUM VITAE

Name Surname: Alper Ulu

Place and Date of Birth: Van, Turkey - 18/05/1990

E-Mail:ulua@itu.edu.tr

B.Sc.:Yıldız Technical University

M.Sc.: Istanbul Technical University

Professional Experience and Rewards:

October 2013 - ...	Product Development Engineer, Argela Technologies
July 2012 - October 2013	Software Engineer, Netas Inc.

PUBLICATIONS/PRESENTATIONS ON THE THESIS

- A. Ulu and H. K. Ekenel, "Convolutional neural network-based representation for person re-identification," 2016 24th Signal Processing and Communication Application Conference (SIU), Zonguldak, 2016, pp. 945-948.