AN END-TO-END CONVOLUTIONAL NEURAL NETWORK FRAMEWORK FOR LOW-RESOLUTION ATTRIBUTE RECOGNITION

DÜŞÜK ÇÖZÜNÜRLÜKLÜ ÖZELLİK TANIMA İÇİN UÇTAN UCA EVRİŞİMLİ SİNİR AĞI ÇERÇEVESİ

RAMIN ABBASZADI

ASSOC. PROF. DR. NAZLI İKİZLER CİNBİŞ

Supervisor

Submitted to Graduate School of Science and Engineering of Hacettepe University

as a Partial Fulfillment to the Requirements

for the Award of the Degree of Doctor of Philosophy

in Computer Engineering

December 2022

ABSTRACT

AN END-TO-END CONVOLUTIONAL NEURAL NETWORK FRAMEWORK FOR LOW-RESOLUTION ATTRIBUTE RECOGNITION

Ramin ABBASZADI

Doctor of Philosophy, Computer Engineering

Supervisor: Assoc. Prof. Dr. Nazlı İKİZLER CİNBİŞ

December 2022, 82 pages

In video surveillance, visual person attributes such as gender, backpack, and type of clothing are crucial for searching and re-identification. For detecting and retrieving these attributes with high accuracy, the availability of high-quality videos is a necessity in general. The details in an image are described by image resolution; the higher the resolution, the more image details. However, in real-world video surveillance systems, videos are usually captured from a far distance, resulting in low-resolution person regions. The technique used for solving this obstacle is super-resolution, which constructs high-resolution images from several observed Low-Resolution images or one single Low-Resolution image. This thesis examines this problem and proposes an end-to-end Convolutional Neural Network that combines a Super Resolution network and Multi-Attribute detection network for more effective Multi-Attribute detection.

Our framework consists of joint training of two main parts, the Super-Resolution and attributes learning. We use different Super-Resolution algorithms in the first part of the proposed method. For this purpose, some well-known and high-quality Super-Resolution algorithms were tested, and finally, two methods entitled EDSR and DBPN were selected. We evaluate the proposed method on two benchmark datasets, Market-1051 and DukeMTMC-reID, labeled with some important labels (attributes) and predict every image label. Experimental results on these two benchmark datasets demonstrate the effectiveness of the proposed approach for the Low-Resolution multiple attribute learning task. Furthermore, we also propose a higher-level linear combination scheme of the two network types (with and without super-resolution), yielding superior results in person attribute recognition.

Keywords: super resolution, multi attribute learning, convolutional neural networks, person attribute recognition, low-resolution recognition.

ÖZET

DÜŞÜK ÇÖZÜNÜRLÜKLÜ ÖZELLİK TANIMA İÇİN UÇTAN UCA EVRİŞİMLİ SİNİR AĞI ÇERÇEVESİ

Ramin ABBASZADI

Doktora, Bilgisayar Mühendisliği

Danışman: Doç. Dr. Nazlı İKİZLER CİNBİŞ

December 2022, 82 sayfa

Video izlemede cinsiyet, sırt çantası, kıyafet türü gibi kişisel görsel özellikler, kişi arama ve / veya yeniden kimlik tespiti için çok önemlidir. Bu öznitelikleri yüksek doğruluk oranıyla tespit etmek ve geri almak için, yüksek kaliteli videoların mevcudiyeti genel olarak bir gerekliliktir. Görüntüdeki detaylar görüntü çözünürlüğüyle tanımlanır, çözünürlük arttıkça detaylar da artar yani doğru orantılılardır. Ancak, gerçek dünyadaki video gözetim sistemlerinde, videolar genellikle uzak mesafelerden yakalanır ve bu da kişilerin bulunduğu bölgelerin düşük çözünürlüklü olmasına neden olur. Bu sorunu çözmek için kullanılan teknik, gözlemlenen bir veya birkaç düşük çözünürlüklü görüntüden yüksek çözünürlüklü görüntüler oluşturan Süper-Çözünürlüktür. Bu tezde, bu soruna bakıyoruz ve daha etkili Çok-Özellikli algılama için Süper-Çözünürlük ağını ve Çok-Özellikli algılama ağını bir araya getiren uçtan uca Evrişimli Sinir Ağları kullanmayı öneriyoruz. Çerçevemiz, iki ana kısmın, Süper-Çözünürlük ve öznitelik öğrenme kısımlarının ortak eğitiminden oluşur.Önerilen yöntemin ilk bölümünde farklı Süper Çözünürlük algoritmaları kullanıyoruz. Bu amaçla, bazı iyi bilinen ve kaliteli Süper Çözünürlük algoritmaları test edilmiş ve son olarak EDSR ve DBPN adlı iki yöntem seçilmiştir. Bu tez, önerilen metodu önemli özniteliklerle etiketlenmiş ver her görüntü etiketini tahmin edebilmeyi sağlayan Market-1051 ve DukeMTMC-reID

veri setleriyle değerlendiriyor. Bu iki kıyaslamalı veri setlerindeki deneysel sonuçlara göre veri setleri, düşük çözünürlüklü Çok-özellikli öğrenme metodu için önerilen yaklaşımın etkililiğini açıklıyor. Ayrıca, kişi öznitelik tanımada üstün sonuçlara ulaşılmasını sağlayan daha yüksek seviyeli iki ağ tipinin (Süper-Çözünürlüklü ya da Süper-Çözünürlüksüz) lineer kombinasyon taslaklarını da bu tezde inceleyip sunuyoruz.

Anahtar Kelimeler: Süper-Çözünürlük, Çok-Öznitelikli öğrenme, evrişimli sinir ağları, kişi öznitelik tanıma, düşük çözünürlüklü tanıma.

ACKNOWLEDGEMENTS

First and foremost, I would like to thank to my supervisor Assos. Prof. Dr. Nazlı İKİZLER CİNBİŞ for her guidance, support, encouragements and toleration throughout the preparation of this thesis.

Furthermore, I would like to thank my thesis committee members, Assos. Prof. Dr. Erkut ERDEM, and Asst. Prof. Dr. Emre AKBAŞ for reviewing my thesis and their valuable comments.

I would like to thank my friend Dr. Ali NEHRANI, for his guidance and support.

I sincerely appreciate my mother, SEDDIGHE, my late father, FERIDOUN, and my three brothers (SABER, NASER, and AMIR), who wholeheartedly supported me on my education path. Although words do not suffice to thank them enough, I will never give up on learning, especially in memory of my honorable father.

Finally, I would like to thank my children PARSA, and SETAYESH.

CONTENTS

Page

TABLES

FIGURES

ABBREVIATIONS

PAR	:	**P**erson **A**ttribute **R**ecognition
MAR	:	**M**ulti **A**ttribute **R**ecognition
MCR	:	**M**ulti **C**lass **R**ecognition
CV	:	**C**omputer **V**ision
AI	:	**A**rtificial **I**ntelligence
ML	:	**M**achine **L**earning
SVM	:	**S**upport **V**ector **M**achine
SR	:	**S**uper **R**esolution
LR	:	**L**ow **R**esolution
SISR	:	**S**ingle **I**mage **S**uper **R**esolution
MR	:	**M**iddle **R**esolution
HR	:	**H**igh **R**esolution
CNN	:	**C**onvolutional **N**eural **N**etwork
MLCNN	:	**M**ulti **L**abel **C**onvolutional **N**eural **N**etwork
DL	:	**D**eep **L**earning
DBPN	:	**D**eep **B**ack **P**rojection **N**etwork
EDSR	:	**E**nhanced **D**eep **S**uper **R**esolution
SRMAR	:	**S**uper **R**esolution **M**ulti **A**ttribute **R**ecognition
SRMAR-D	:	**SRMAR** network uses **DBPN** as backbone
SRMAR-E	:	**SRMAR** network uses **EDSR** as backbone

1. INTRODUCTION

1.1. Motivation

Vision technology, both in hardware and software, has improved dramatically in recent decades, and it will take the place of other sensors in various applications. Nowadays, micro-controllers can run vision algorithms and process vision data. As a result, creating a vision-based system has now become much more accessible than a decade ago, allowing us to develop applications, devices, and robots based on vision systems much cheaper. The question ”Why is visual sensing prevalent?” can be answered by describing its applicability in the industry of any type, in research, Etc. Visualizing visual data is better for humans to understand and interpret what is going on in the environment. Fortunately, visual processing systems are working very well in many areas, and the applications are growing yearly.

Starting from automation in industries, we can easily understand its importance in the automation of repeating processes and tasks to complex ones, including navigation, all depend on vision-based systems. One of the areas in that vision and visual sensing has an essential role in the development and automation is the surveillance and security sector. Due to the importance of automation in this sector, considering the cost of labor, dangers that human operators are facing, and many other reasons, considerable research has been conducted in this area, and still, the number is growing. Therefore, vision-based automation in surveillance systems is the core technology to address many challenges.

One of the critical challenges in surveillance systems is scene understanding, recognizing possible dangerous activities, recognizing humans in the scene, objects, Etc. (see Figure 1.1. as an example). The recognition task is based on the data from the camera as a base vision data streamer. The term data is very general; at the end of the day, the algorithms are working on only a tiny percent of all data collected. Data come with noises, and a large part of the data is useless, and all should be processed in real-time, especially in surveillance systems.

Re-identifying humans based on images and videos is a fundamental and significant problem that is very important in designing surveillance systems. Currently, there is much research

Figure 1.1. Prevention of crime and terrorist incidents and threats is one of the goals of visual surveillance systems. This image shows a hypothetical visual surveillance system used with a human system that can recognize and control faces from a given blacklist and raise an alert when a match is found. [3].

on the subjects, and recently is improved significantly, and it was expected to surpass human performance in the next couple of years.

To re-identify a person, parsing a human from a 2D image, relies only on the visual appearance of the human. One approach is quantifying the appearance of being able to annotate data to taking the advantage of machine learning algorithms as well as being able to study the improvements in the area by assigning attributes to humans in the scene.

Person Attribute Recognition (PAR) is a recently evolving task in computer vision that is of particular interest, especially for visual surveillance systems. The PAR task consists of recognizing personal characteristics such as gender, age, clothing style, hair, and many other attributes. Several challenges are associated with this task, including varying illumination conditions, different viewpoints, and low resolution due to far distances. Furthermore, this task requires analysis of finer details; when the details are lacking due to Low-Resolution (LR), the PAR becomes very difficult. Therefore, the best technique for solving LR images and videos is Super-Resolution (SR). SR is a set of algorithms or methods of up-scaling videos or pictures. The SR methods and techniques principles are similar: creating one

up-sized image from several different images or estimating a high-resolution (HR) image from its LR image. However, the image resolution is limited by the hardware and capturing systems, such as image sensors (for example, CCD) and optics. Constructing optical components and imaging chips to capture HR images is expensive and not practical in most real cases. With breakthroughs in computer vision research, SR received substantial attention with more practical aspects.

In traditional machine vision approaches, handcrafted features (e.g., Color Histograms, Local Binary Patterns, Histogram of Oriented Gradient, Bag of Visual Words, and so on) are the methods used for features or attributes representation that is followed by classification with a standard classifier such as the trendy Support Vector Machine (SVM) [21]. Recently, in contrast to handcraft features, the learned features with deep learning structures like Convolutional Neural Networks (CNNs) have shown great potential for different vision tasks. For example, multi-attribute networks show great potential in understanding the involved human scene, which can be a forward step for human activity recognition. This area has been active in recent years due to its potential tremendous application opportunities [6, 22–24].

According to our experience, Multi-Attribute (MA) model performance on LR images does not improve even when training on LR data because of information deficiency in such images. Considering that the network is training with a classification loss function, it may not solve the LR issue independently. Our idea is to improve the images by applying SR networks that improve image resolution. For this purpose, we first use the SR network separately on LR images and produce HR images. Then, using these HR images as input to the MA network led us to better accuracy results. Therefore after getting these results, we combine the SR and MA networks as an end-to-end network to get exact results. Finally, our results show that the accuracy of a combined SR network with an MA network dramatically improves, supporting our hypothesis. In this thesis, we try to solve the problem of Multi-Attribute Recognition (MAR) and Multi-Class Recognition (MCR) for LR 2D images. In this dissertation, we propose an end-to-end learning model by merging the power of SR and MAR models to get more accurate recognition. Because the higher image resolution makes better accuracy in the MAR system, we propose a merged model of the SR network with a

multi-attribute learning network. We establish experiments on two benchmarks, the Market-1501 [1], and the DukeMMTC-reID dataset [2]. As a result, we prove from experimental results that the accuracy of the MAR network increases by merging it with SR networks. Thus, we prepare an end-to-end neural network model that starts with an SR network and ends with a MA network, we call this end-to-end CNN architecture as the super-resolution multi-attribute recognition (SRMAR) model. To the best of our knowledge, this research is the first end-to-end learning model for person MAR, which proposes using extra information (HR images) to heal the information loss LR images. We further propose a linear combination of the SRMAR network and the MAR network to boost recognition performance. Our experiments verify that combining models is better for the LR person to attribute recognition.

1.2. Contributions of the Thesis

In summary, our main contributions in this thesis are:

- It is shown that using the SR as a prepossessing in LR images increases the accuracy of multi-attribute multi-class classification recognition.
- In this thesis, we make a combined CNN architecture called SRMAR that uses an SR network trained to reconstruct LR images and a MAR network.
- The linear combination scheme combines the SRMAR model with the MAR model to get a better mean accuracy of attributes proposed in this thesis.
- We improve the state-of-the-art in PAR even on LR images.
- We extensively tested the proposed models on two benchmark datasets widely used for PAR, and there is a reproducible state-of-the-art result.

1.3. Organization of the Thesis

This dissertation shows how a super-resolution algorithm can increase multi-attribution accuracy in low-resolution images. The structure and rest of this thesis are as follows:

- Chapter 2 summarizes the background of the methods and techniques used in this thesis. Convolution Neural networks (CNNs) and some of the most famous and common CNN architectures are summarized in this chapter.

- In chapter 3, a review of the literature on Multi-Attribute Recognition and Super-Resolution methods is given, and some new articles about them are reviewed.

- Chapter 4 gives the details of our proposed methodology. In addition, details of using our combination model and linear combination model are given in this chapter.

- Chapter 5 presents the implementation details and the experiments based on the proposed methods, and this chapter also describes using two famous and standard datasets. Finally, a comprehensive evaluation of the methods and a comparison of results are presented.

- Finally, in Chapter 6, we discuss the conclusion of this work and possible future research directions.

2. BACKGROUND

This thesis focuses on increasing the multi-attribute recognition accuracy of low-resolution person images using super-resolution algorithms. More specifically, it aims to increase the multi-attribute recognition accuracy of low-resolution images by combining the super-resolution network with the multi-attribute network as the end-to-end network. This chapter summarizes some general definitions, methods, and techniques used in this dissertation.

2.1. Visual Attributes

As discussed in the introduction chapter, vision-based automation of surveillance is taking 2D videos, and all processes and decisions are based on the person's visual appearance in the image. Related semantics, auxiliaries, or higher level features can be applied to understandably pars this visual data. Defining attributes based on the appearance of the human is essential in collecting and annotating data for the supervision of algorithms and making the problem interpretable. For example, a car has a type, color, size, Etc., which is visually recognizable. For humans, clothes, wearing style (e.g., a spotted skirt, rather than just any skirt), hair (color, size), carrying something, and even the clothes' color could be considered recognizable attributes. These visually recognizable attributes are what we will refer them as visual attributes throughout the thesis. The visual attributes are selected based on available authentic datasets.

2.2. Human Visual Attributes

Human vision attributes are those semantics that a human can recognize different humans from each other or a single human in different scenes. Based on this definition, we may be able to list many attributes, but in computer vision, only recognizable attributes (or partially recognizable) based on their occurrence in the data (i.e., 2D video) are listed. For example, it

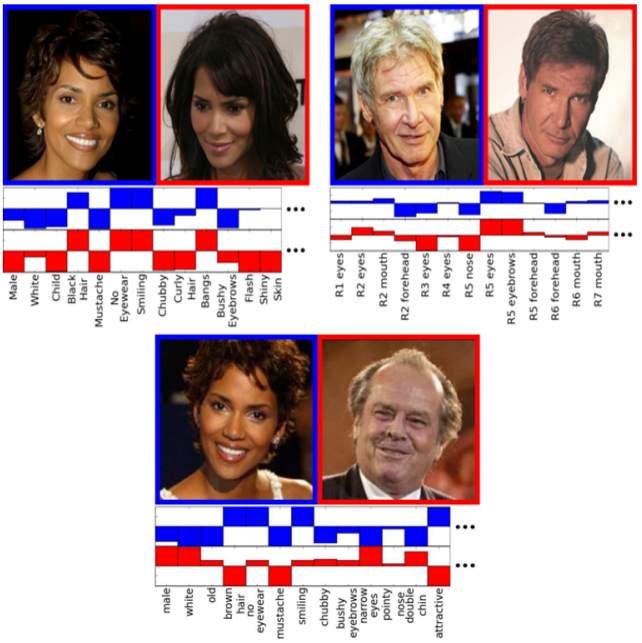

Figure 2.1. Sample of face verification with visual attributes [4]. The first and second row pictures show attribute values of the same and different persons.

immediately can be understood that the human head/face [1], clothing (wearing a hat, skirt, the t-shirt is visually recognizable), clothing colors (both upper body and lower body), hairstyle (having hair (or not), hair color, long or short, smooth or curly, Etc.), the status of carrying something that a person commonly does (backpack, hand back, Etc.). Due to the importance of the definition of human visual attributes, the subject is studied extensively, and based on that, some valuable datasets were created [1, 2, 22]. There are common attributes in different datasets and differences that come from the applications for which a particular dataset is prepared. Generally, datasets consider 5 to 15 different human visual attributes. The number of attributes is not as crucial as their descriptiveness of them. However, a particular attribute must become informative enough to contribute to the recognition task. Using these attributes is so effective in visual tasks. For example, as shown in Figure 2.2., based on the human visual attribute as a search query [5] to retrieve a face.

[1]Face recognition is another crucial topic in computer vision where the only human face is taken into account (as shown in Figure 2.1.)

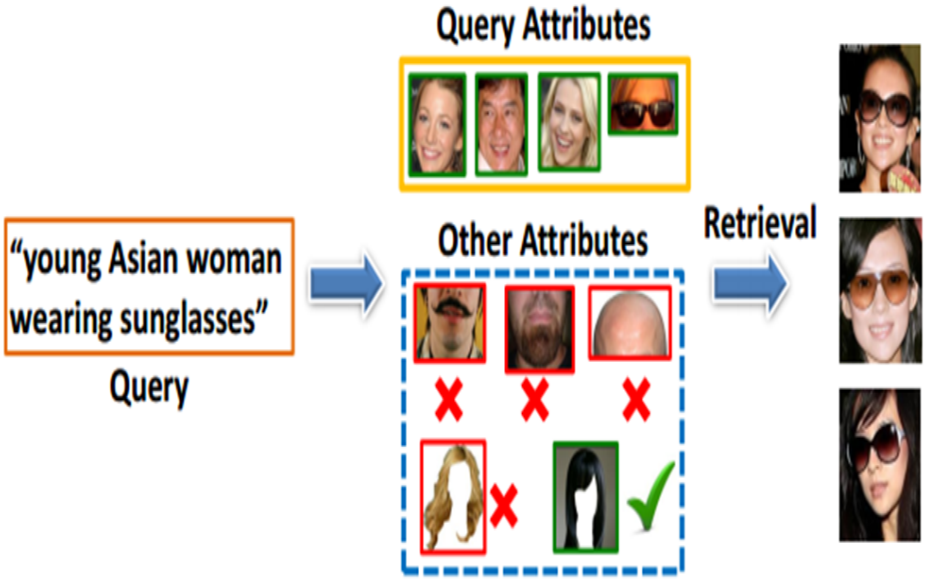

Figure 2.2. Sample of image retrieval and searching with Multi-Attribute [5], in addition to considering given multi-attribute keywords inside the given query, this research also considers the remaining attribute that is not part of the given query. For example, in this Figure target is retrieving Asian women with sunglasses. In their algorithm, not only are multi-attributes of the given query considered, but also some other attributes related to given multi-attributes are controlled; for example, in this sample query, finding results should not have a mustache, blond hair, or beard.

2.3. Person Attribute Recognition

Person attribute recognition (PAR) recognizes whether a person poses a certain attribute from the given image. As already discussed, based on the status of the attributes, a person can be recognized by others in one or multiple scenes based on the human visual attributes defined in the previous section. As it is clear, PAR is a sub-task relative to person recognition, a high-level semantic recognition of a person's image. Figure 2.3. includes an example of person multi attributes.

2.4. Resolution

The resolution, as it is clear from the meaning in our case, is the number of pixels that represent a visual scene. In 2D images, like the data that is considered in the thesis, $w \times h$ pixels color per channel representing the whole scene w for width and h for height. It is

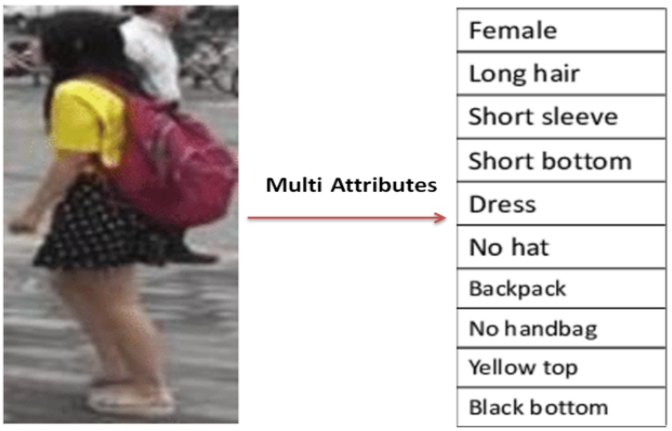

Figure 2.3. Example of person multi attribute prediction [6].

not needed to mention that the number of pixels is very important in computer vision and in general in any visual process. Imagine a scene that is represented with 4 pixels and the same scene by 4000 pixels. The more pixels, the more details can be stored in the 2D image. However, the problem is not straightforward as that, for example storing and computation load can become exponentially costly for the higher number of pixels, as a result, based on the specific task, the number of pixels should be determined so that helps us to reduce computation and storage cost in one side and do not affect the recognition performance from the other side. Image resolution is a core keyword in this thesis. We will work on data with the resolutions 8×16, 16×32, 21×42, and 32×64. With these resolutions, attribute recognition is very challenging even for humans.

2.5. Super-Resolution

Super-Resolution (SR) is a set of algorithms or neural network models that upscale the size of the input image. In classic computer vision, for upsampling an image, the information for the new pixel is taken from the neighbor pixels; however, in super-resolution, the neural network learns information about the new pixels from all the pixels, from the context, shape, or even from other images. In theory, this makes the SR neural network very powerful in the upsampling task (Figure 2.4.). SR neural networks' power and capability can help us address very challenging upsampling tasks. In this thesis, we tried to understand how helpful

SR models are in multi-attribute recognition datasets. For example, the SR model can take a very low-resolution image of a person, upsample it, and the output can be fed to another neural network to recognize, for example, a person's attribute. A comparative study reveals that SR models are capable of generalizing on unseen data.

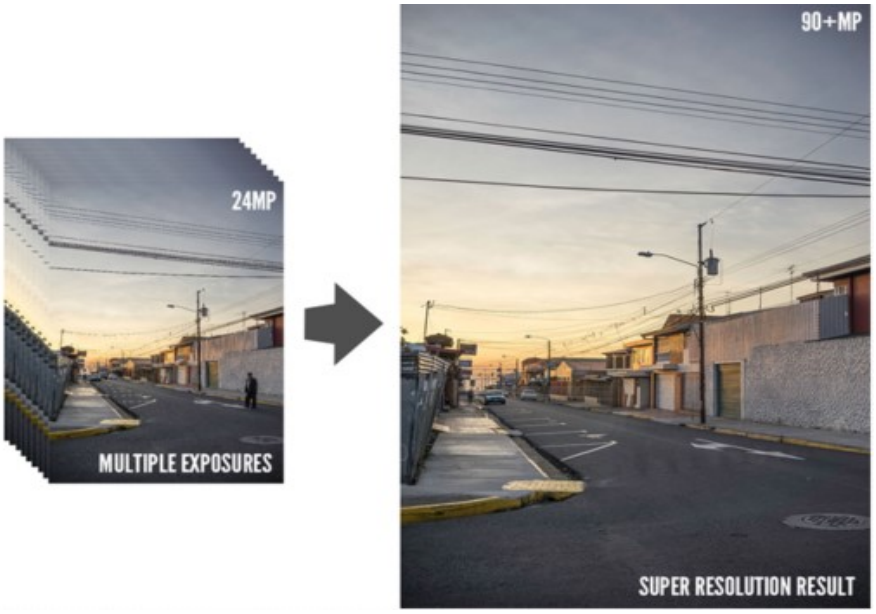

Figure 2.4. Creating one up-sized image from several different images or pictures [7].

2.6. Neural Networks

Neural networks are essential in deep learning algorithms and are a subset of machine learning. The structure and name of neural networks inspire the human brain. The neural networks, also known as artificial neural networks (ANNs), are included in node layers. These layers have an input, one or more hidden, and output layers. Each node is connected to another node and correlated with weight and threshold. With controlling entries like thresholds, activating the node for sending or not sending data to other nodes becomes obvious. Because of access to many different input types, such as images, videos, sounds, and files, many problems are solved with neural networks. For instance, we can refer to some of these applications as pattern recognition, self-driving, face recognition and detection, image classification, data mining, medical diagnosis, spam filtering, and more. There are many types

of Neural Networks. Some important are Feed Forward Neural Networks (FNNs), Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and LSTM Neural Network. In this dissertation, generally, we use Convolutional Neural Network.

2.7. Convolutional Neural Networks (CNNs)

One of the famous types of artificial neural networks is Convolution Neural Networks (CNNs) which have incredible results in subjects like computer vision, natural language processing, and other purposes [25–29]. The CNN was initially introduced by [30] and became favored and popular with the release of AlexNet architecture [8]. Using CNN with AlexNet architecture points to achieving enormous success on the ImageNet dataset. The structure of their network is shown in Figure 2.5. A CNN architecture, like an ANN, is a sequence of layers. Each layer takes the input, performs a transformation, and passes it to the next layer through various functions and methods. The following subsections summarize some essential parts of this type of neural network.

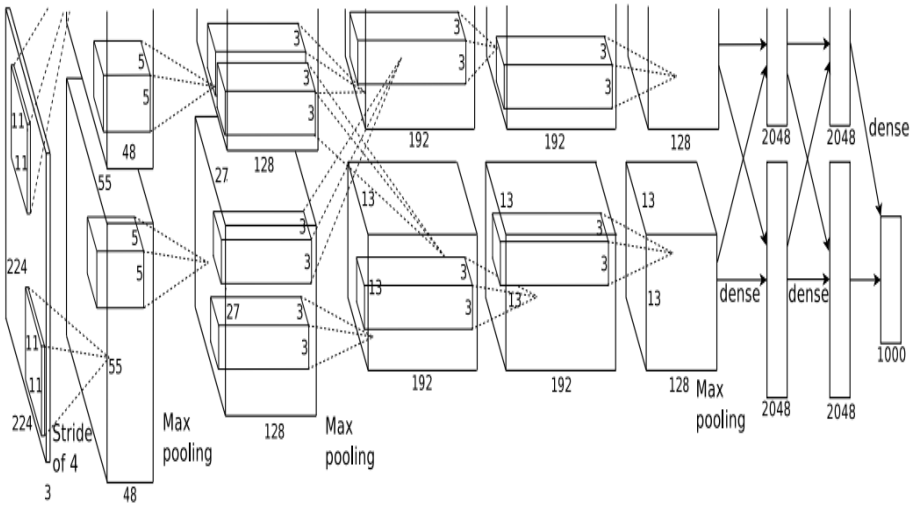

Figure 2.5. AlexNet network structure [8]

2.7.1. Convolutional Layers

One of the base parts of CNN architecture is the convolutional layers which the extraction features from the input are the duty of these layers. For this purpose, some linear and none linear functions or combining of both of them are used by convolutional layers. As a mathematical aspect, convolution act like a linear function (element-wise product) on a tensor (input) that is the array of numbers and a small array of numbers called a kernel.

2.7.2. Pooling Layers

Another layer of CNN is pooling layers, also known as the down-sampling function; it reduces the number of parameters in the input. As an operation, it is similar to the convolution layer that sweeps a filter across the entire input but, unlike the convolution layer, does not have any weights. While much information is lost in the pooling layer, it also benefits. They help improve efficiency, reduce complexity, and limit the risk of over-fitting. Max pooling and Average pooling are the two main types of pooling.

2.7.3. ReLU Correction Activation Function

Rectified Linear Units known as RelU refer to the real non-linear function defined by $ReLU(x) = max(0, x)$. This layer output must be positive. Therefore it replaces all inputs with negative values by zeros and acts as an activation function.

2.7.4. Fully-Connected Layers

The fully-Connected layer is used at the end of CNN to classify the inputs based on the features extracted through the previous layers and filters. The name of the fully connected describes itself and refers to every neuron from the output layer that connects to every neuron's previous layer. It takes their output and flattens and sends them to a one-dimensional array or vector of numbers.

2.8. CNN Architectures

The minimal architecture of a CNN includes a few Convolution layers (Conv) as initial layers and activation functions such as Rectified Linear Unit function, followed by pool layers. This pattern repeats until the input image gets spatially tiny enough. Finally, to classify the input, a few Fully-Connected (FC) layers are used at the end of this architecture. Some of the most common ones are LeNet[31], AlexNet [8], VGGNet [32], GoogleNet [33], ResNet [34], and DenseNet[35].

2.8.1. LeNet

LeNet is one of the earliest CNN proposed by Yann *et al.* [31] in 1998. This network is used for recognizing handwritten and machine-printed characters. The first architecture of LeNet has five layers with learnable parameters. It has two fully connected layers after the three sets of convolution layers and average pooling layers. The input image to this model is a 32×32 grayscale.

2.8.2. AlexNet

The AlexNet is one of the famous CNN [8] that includes eight layers. The first five are CNN layers, starting from an 11x11 kernel, and the last three layers are Fully-Connected. This CNN in Visual Recognition Challenge (ILSVRC) is one of the winners of ImageNet Large Scale [36] in 2012. The first architecture includes max-pooling layers, ReLu activation functions, and dropout for the three enormous linear layers. The used architecture of AlexNet [8] is shown in Figure 2.5.

2.8.3. VGGNet

Another famous CNN that we talk about it is VGGNet. This CNN was proposed by Simonyan et al. [32]. Discussing the effect of CNN depth on its accuracy is the main contribution of their research. Originally fixed-size RGB images (224x224) are passed through a stack of convolutional layers. Then, three FC layers follow this stack of convolutional layers. The first two FC layers have 4096 channels, and the third one contains 1000 channels equal to the number of classes in ImageNet.

2.8.4. GoogleNet

GoogleNet or Inception V1 that proposed by a research at Google in 2014 [33]. This model uses new techniques such as 1x1 convolutions in the middle of the architecture to decrease parameters. By reducing the parameters, they also increase the depth of the architecture.

2.8.5. ResNet

After proving that CNNs are very successful in classification and solving other artificial tasks, residual deep networks like ResNet [34] were introduced as one of the highest accuracies in image classification problems. Furthermore, it shows that CNN can increase layers to hundreds or thousands during training without adverse effects on performance, which is possible because of the residual learning algorithm.

2.8.6. DenseNet

Another type of convolution neural network is DenseNet [35] which uses the layers entitled with the dense connection for connection between layers. This connection layer uses a linear operation that connects every input to every output by weight which layer's connection happens during a feed-forward fashion.

2.9. Fine Tuning and Transfer Learning

Every neural networks and CNNs have a task and a purpose; for example, in a face detection and recognition system, the target is to find and identify the faces in videos or images. Alternatively, in classification systems, the target of used CNN is the classifier of objects. Of course, we can refer to many challenging problems that need more time and human resources, but artificial systems, especially CNN's and neural networks, can solve them quickly; the question here is, how can CNN's and neural networks do this? Training a CNN is critical based on using the best parameters during training and using tremendous related data to its task. Choosing the best parameters requires more experiments and studies, but gathering suitable datasets and labeling those is challenging and consuming time. Sometimes, we have to reproduce data to balance datasets and avoid network over-fitting because of a lack of suitable datasets. This technique is known as data augmentation. Another choice is to use an already trained model and train it some more rather than training from scratch. Fine Tuning and Transfer Learning are the terms that increase CNN accuracy results. In fine-tuning, we freeze some early layers instead of training in all layers and continue training only at the final layers. In Transfer Learning techniques, the pre-trained model is used as a learned model and utilized to train some new category or objects in an image that has not been trained before.

2.10. End-To-End Learning

There are essential concepts defined as non-end-to-end and end-to-end in deep learning subjects. Both describe the learning approach at the highest level: end-to-end means mapping the input directly to the final output. In this method, a function of weights is learned across back propagation from a signal that traces back. The end-to-end method is the preferred means of training nowadays, as the most exciting features for the target metrics are learned via optimization. On the contrary, non-end-to-end does not learn a mapping from the input space to the output but depends on intermediate steps that are often tuned by human design and, hence, likely not optimal.

3. RELATED WORK

This section briefly reviews multi-attribute recognition (MAR), and super-resolution (SR) techniques.

3.1. Multi Attribute Recognition

The study of person attributes or visual recognition of pedestrian attributes, such as clothing color, clothing style, age, gender, and attributes like those, has recently been in high demand at CV and AI systems. However, multi-attribute modeling for CV tasks is relatively contemporary and became extensively employed in the following years, first proposed by Ferrari and Zisserman [37]. The common point of all of these studies with attention to human recognition concepts is combining an ensemble selected and more important visual attributes to some ML detectors and classifiers. These attributes also have essential roles in image searching and retrieving. Pioneering works primarily focus on finding feature vector and classifier scores; for example, in work [4] related to face verification, the writers for face retrieving try to combine binary attribute and classifiers outputs. For this purpose, they define some labels to describe the visual appearance of an image or a face. These labels are entitled ”describable visual attributes.”

In another work [38], which is the first surveillance system based on video input, related researchers try to image retrieval by using attributes. In their algorithm, the person’s body attributes are also considered besides controlling face and facial attributes. To learn these attributes, they used a large training dataset and a standard ML algorithm, ”Adaboost Classifier with Haar features,” then. For extracting facial attributes, a detector ”Viola-Jons” was trained. For an experiment, they took 9800 frontal face images from the Labeled Faces in the Wild [39] dataset.

Research on methods that exploit objects’ semantic attributes has revived the attention of the CV community. However, in more initial research about MA queries, the researchers try to train and classify attributes independently and then combine scores of those they were trying

to find the best searching or retrieving images based on that given query. Studying attributes in an as independent manner is not the most efficient way always. For example, Seddiquie *et al.* [5] exploited a wise pair relationship between facial attributes and up and down body attributes. This relationship between different body parts increases and improve search based on MA queries. In their proposed framework, for retrieving images from a given query, both attributes are included in search sentences and attributes that make more information about the given query are used. In evaluating performance, they chose two methods: Reverse Multi-Labeling [40] and TagProp [41]. Regarding ranking, they compared their ranking model with rankSVM [42], rankBoost [43], and DORM [44].

One initial research about re-identification with mid-level attributes was proposed with Layne *et al.* [45]. In this work, firstly, the picture of the person is divided into six equal parts horizontally for extracting these mid-level attributes. Then texture and color features of these parts are extracted and trained with an SVM detector. For validating their model, two challenging datasets (VIPeR [46], and i-LIDS [47]) are selected.

One of the obstacles in many real-world surveillance scenarios is that pictures are taken from the person or pedestrian's face or body, usually unclear or occluded or taken far from a distance. This problem makes two fundamental challenges known as appearance diversity and appearance ambiguity in attribute results at a far distance. Therefore some research to solve these problems was published. Deng *et al.* [22] firstly, create a new dataset entitled PETA that is taken from another ten famous person datasets for more variety in attributes. For experimentation and preset benchmark results in PETA, they evaluate the performance of their algorithm with SVM with intersection kernel [48] and Markov Random Field by choosing two kernels(Gaussian and Random Forest). Then, they randomly split the dataset images into 9500 for training, 7600 for testing, and 1900 for verification. In this study, the 35 most important attributes are chosen. For extraction features of images, low-level and texture features are used.

In multi-camera scenarios, exploiting and studying shared attributes and information is essential for increasing re-identification accuracy in MA systems. In addition, finding a correlation between attributes is very helpful for these scenarios because some attributes frequently co-occur. For example, the attribute of baldness is likely to be highly correlated to males rather than females. Chi *et al*. [49] proposed an algorithm that simulates the correlation between attributes by using a features vector in the same person at the multi-camera. For utilizing correlation between features, they use MTL[50] algorithm. For evaluation, they used 4 public and famous datasets iLIDS-VID [51], PRID [52] and VIPeR [46] and SAIVT-SoftBio [53] which in all of them included different person pictures from multi-camera. They also used color and texture features and SVM for feature extraction and detector, which were routine in those days.

Recognizing the gender of an interested person is easy and primarily accurate for humans, whether just part of the person is visible or in arbitrary pose positions. For example, if we see a person's lower body, with the clothing style, gender is distinguished by our brain. Alternatively, if the target is to recognize persons that have an attribute like a hat, recognizing this attribute in the top part of the person is more straightforward than recognizing it in the total image of the complete body of the person; therefore, the ability of divided body parts and recognizing exciting attributes in the related parts is an essential role in some research. One of those works is Lubomir *et al.* research [54]. They use different keys related to the viewpoints and pose to recognize different attributes. The challenge in their method is that the system must be detected and align the parts well. Their algorithm included three parts. Prediction of attributes based on pose let types are done in the first part, and a combined result of these attributes values is obtained from the second part. Finally, the correlation between different attributes is obtained in the third part.

All methods and researches discussed in previous sections for attribute recognition have two essential weaknesses, primarily handcrafted features like a color histogram and local binary patterns, and some ML algorithms are used for feature extraction. These features in real video surveillance scenarios cannot handle all requests. Secondly, the correlation between the person or pedestrian attributes is mainly missed in these researches. As a result, to solve

these obstacles, the sense of using new methods and algorithms is more required. One of the recent technologies that attracted more attention for researchers at that time was neural networks. The ability of NNs to solve some challenging problems was apparent to researchers. However, research about MAR was not deprived of this new technology; therefore, problems with handcrafted features and correlation between multi attributes mainly were solved with this new technology. One of the ancestor research about this is Dangwei *et al.* [23] research. With attention to NNs' abilities, they use two deep learning models to solve recent drawbacks. One model is entitled DeepSAR for recognizing each attribute, and another is entitled DeepMAR for detecting relationships between attributes. DeepSAR model is finetuned based on CaffeNet [55], which is the same as AlexNet [8], but the order of the normalizing layer and pooling layer changed. They first evaluate their models on PETA [22] dataset. DeepMAR has been evaluated on APIs [56] dataset to verify their method further.

Li *et al*. [23] proposed two models based on deep architectures to address hand-crafted features' drawbacks and ignore the relationship between attributes. Their single attribute recognizing model (DeepSAR) is designed for recognizing each attribute individually. The proposed second model [23] is the deep learning framework to exploit the relationship between attributes.

Using MLCNN for studying and predicting multiple attributes together had a special place and importance in the years between 2015 and 2016. One research about this is Jianqing *et al.* [24] research. They use body parts of a pedestrian image as inputs to MLCNN and filter independently. For evaluation, they used VIPeR [46] and GRID [57] datasets.

Another work that considers dependencies between attributes is a work proposed by Patrick *et al.* [58]. They train a CNN by considering all attributes together. The base of their CNN is CaffeNet framework [59] that was pre-trained on ImageNet. Their network started with the CaffeNet structure and ended with the proposed custom loss layers. For evaluation, they use two datasets, HATDB, which was initially published by Sharma *et al.* [60] which is labeled with nine binary attributes, and Berkeley - Attributes of People dataset [54]. In addition,

using these public datasets, They made a new dataset entitled PARSE-27K that was taken from cameras inside the city.

Correct recognition of human attributes remains challenging in some situations like viewpoint variations or occlusion of part of a person or pedestrians, different poses, and illumination effects on images. To solve these problems, Yining *et al.* [61] proposed a model that analyzes more variable parts of the target person as not only an individual but also the persons near an interested person analyzed. They also have a scene-level analysis which helps them get better attribute recognition of the target person with the above challenges. They change Fast R-CNN [62] to help the study of deep hierarchical contexts of images. They evaluate their method on the two datasets, Berkeley-Attributes of People [54] and HAT [60] datasets. The WIDER dataset [60] was introduced with them in this research.

Yutian *et al*. [6] proposed a method and tried enhancing the performance of expansive pedestrian re-identification by using attribute labels. In this research, two subjects, person re-identification and attribute recognition studied. As same to recent research about MAR, their model is based on ResNet-50 [63] and CaffeNet [59] that pre-trained in ImageNet [36] but fine-tuned with new annotated attributes of two datasets entitled The Market1501 dataset [1] , and DukeMTMC-reID dataset [2]. Their model structure included two parts, identification and attribute recognition. Firstly, the feature vector of input images via CNN was extracted. Then, based on extracted feature, a classifier module predicts person attributes.

Shi *et al.*[64], proposed a network with two modules: coarse and fine alignment modules. The first module uses a part detector to locate the body parts and form the candidate attributes; then, in the second module, these attributes are aggregated together via a bilinear-pooling layer. Wu *et al*. [65] propose a parallel model, which consists of intra-attention and inter-attention parts to learn the relationships of images and/or attributes.

3.2. Super Resolution

Super-Resolution (SR) is a set of algorithms or methods of upscaling video or images. The base idea of many SR methods and techniques is the same: creating one upsized image from

several different images or estimating an HR image from its LR image. The image resolution is limited by the hardware and capturing systems, like image sensors (for example, CCD) and optics. Constructing optical components and imaging chips to capture high-resolution images is expensive and impractical in most real applications. With breakthroughs in computer vision research, SR received substantial attention in this community and has many applications. This attention comes from two assumption application areas; helping representation for automatic machine perception and visual information for human interpretation. SR arises in many areas [66], such as:

- **Surveillance video:** freezing frames and region of interest (ROI) for human perception and automatic target recognition (for example, looking at the license plate or trying to recognize a criminal's face).
- **Remote sensing:** improving resolution image with several images taken from the same area.
- **Medical imaging (CT, MRI, and Ultrasound):** By using SR techniques and several images limited in resolution quality, images with enhanced resolution are produced.
- **Video standard conversion:** for example, NTSC video signal to HDTV signal.

Pioneering work in SR was published in 1984 [67], and then the term super-resolution itself appeared around 1990 [68]. One of the first approaches in single-image SR is Yang.*et al.* [69] article. Their work is based on signal processing and mathematics and compares signals with sparse signal representation. Based on research on image statistics, sparse representation for each image patch of the LR input image searched, and then HR output using the coefficients of this representation. After searching image patches on LR and HR and joint training two dictionaries, they can enforce the similarity of sparse representations between LR and HR image patch pairs concerning their dictionaries. Therefore, the sparse term of an LR image patch is involved in the HR image patch dictionary for generating an HR image.

CNN-based super-resolution methods have yielded excellent results on previous handcrafted models. For example, in Wang *et al.* [70], the authors use feed-forward network architecture

and combine the conventional sparse coding model with ingredients of deep learning to get better results.

Another work is Dong *et al.* [12, 71]. In this research, a CNN with three layers was trained, and upscale images with an interpolation algorithm (bicubic) were used as input images. The authors using single-image super-resolution (SISR), propose a deep learning method. An end-to-end mapping between the low/high-resolution images is directly learned by this method. Using a deep convolutional network (CNN) for the mapping method produces the high-resolution from the low-resolution image as the output and input, respectively. Their proposed model Super-Resolution Convolutional Neural Network is entitled SRCNN. This model has several appealing properties like simplicity in design and superior accuracy compared with the state-of-the-art example-based methods. Another property is the fast speed for practical online usage even on a CPU because their model uses fully feed-forward and does not need to solve an optimization problem on usage. Moreover, the last property is with using a more extensive and deeper model, the resolution quality of the network can be further improved. For desiring a high-resolution image, they first up-scaled a single low-resolution image to the desired size using the interpolation method "bicubic," which is the only pre-processing they use. Then, the output of pre-processing is used as a low-resolution image. From the low-resolution image, overlapping patches are extracted; each patch is represented as a high-dimensional vector. Each high-dimensional vector non-linearly maps onto another high-dimensional vector. These mapped vectors represent a high-resolution patch. The result of this operation is the final high-resolution image. For this purpose, all the above high-resolution patch-wise representations are aggregated. The output of this operation is expected to be similar to the ground truth image.

Generative Adversarial Networks (GAN) have been found to have an essential role in SR researches [72–76]. Ledig *et al.* [20] focus on SISR and present a GAN for super image resolution. Despite previous researchers that for producing HR images from multiple images, they will focus just on SISR. This work has focused primarily on minimizing reconstruction error. They present a generative adversarial network (GAN) for SR entitled SRGAN. This work produced SR images with four times upscaling factors for the first time. For reaching

this upscaling, a particular loss function is proposed. Their research is the first intense ResNet architecture that uses the concept of GANs. Their main target is to train a generative function that estimates it's corresponding high-resolution for a given low-resolution input image. For this purpose, the writers prepare a generative network as a feed-forward CNN. At the main skeleton of their very deep generator network are some residual blocks which identical layouts and two convolutional layers with small 3x3 kernels and 64 feature maps followed by batch-normalization layers [77] and Parametric ReLU [78] as the activation function is used. For improving the input image resolution, they use two trained sub-pixel convolution layers as offered by Shi *et al.* [17]. They also prepare a discriminator network to discriminate authentic HR images from generated SR samples. In their experiment, They used three benchmark datasets Set5 [79], Set14 [80], and BSD100, the testing set of BSD300 [81].

Yamanaka *et al.* [9] with using Deep CNN, proposed a SISR model that is faster and better performance of ancestor's SISR models. To improve performance, they used deeper CNN layers. However, as we know, deep models get more computation resources. Therefore, those models are not suitable for use on edge devices. Nevertheless, their proposed model achieves at least ten times lower calculation costs. Furthermore, they proposed a lighter network by optimizing its structure with the current eight deep-learning-based SISI methods, as shown in Figure 3.1. This Figure shows that their network comprises two smaller networks: feature extraction and reconstruction networks. In the first part, unlike previous DL-based models where an up-sampled image was often used as an input, to understand the features efficiently, they used an original image as an input in their model. In the second part, the part that details of the image are reconstructed. Normally, more convolutional layers must be used to improve reconstruction results, increasing computation processes. So they proposed a parallelized CNN structure like [82], which normally has one or more one-layer CNNs. Moreover, this CNN caused to reconstruction process to be more efficient and faster. For training, Berkeley Segmentation [83] and Yang *et al.* [69] datasets were used. In the phase of performance evaluation, SET5 [79] dataset was used.

Lim *et al.* [10] developed an Enhanced Deep SR network (EDSR). Due to optimization by removing unnecessary parts in conventional residual networks, their model has a performance

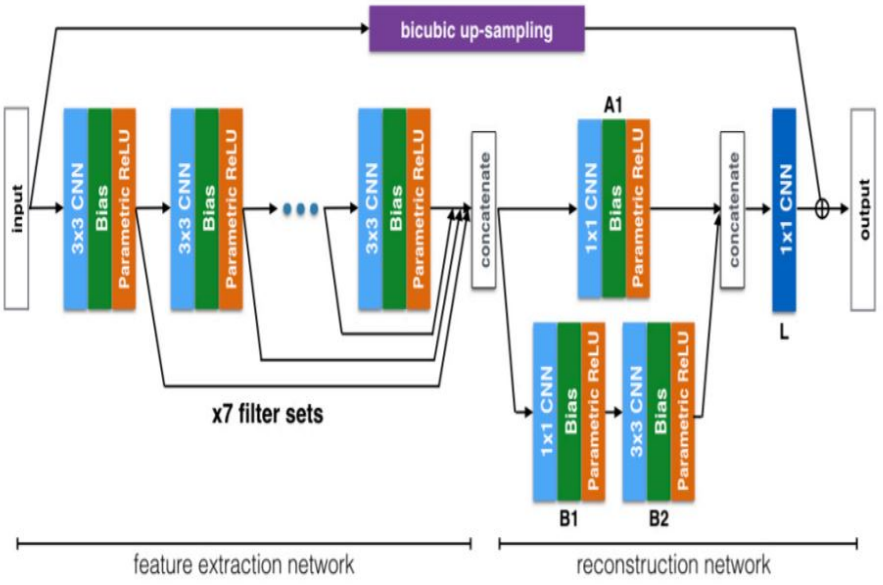

Figure 3.1. The structure of the DCSCN SR network architecture [9]

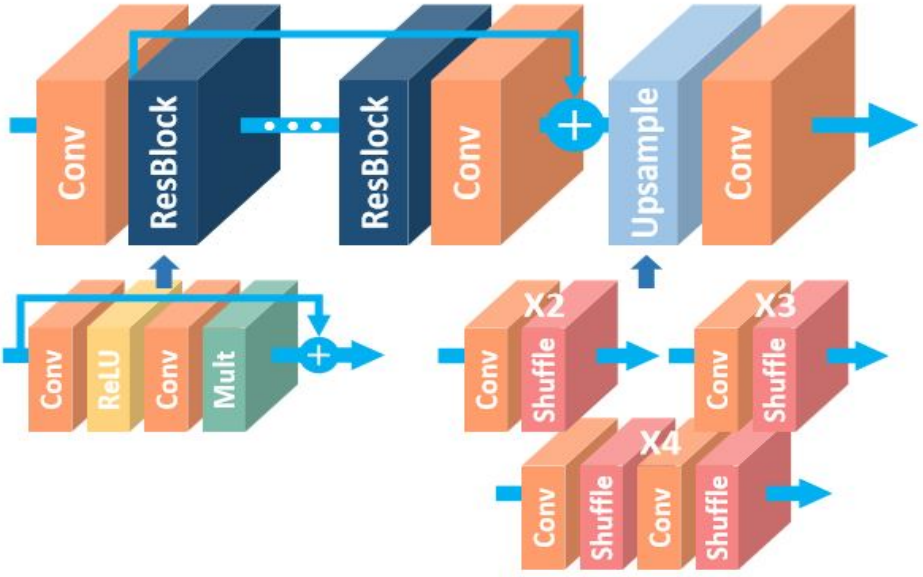

Figure 3.2. The structure of the EDSR single scale architecture [10]

improvement. Their model was further improved by expanding the model size without negatively affecting the training stage. They also proposed a new Multi-Scale Deep SR (MDSR). In their proposed network, they remove batch normalization layers from the network. This action saves around 40% of memory usage during the training procedure compared to the SR algorithm that uses ordinary ResNet in its architecture [20]. They also used a particular

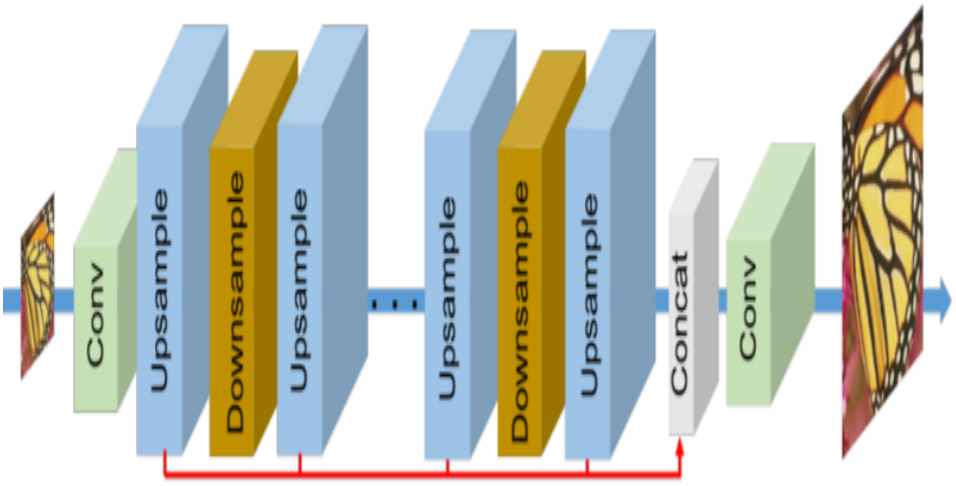

Figure 3.3. Structure of an iterative up and down sampling approach in DBPN SR algorithm. This structure tries to minimize the error between up and down sampling.[11]

strategy in training for factors x3 and x4; they fine-tuned these networks with a pre-trained x2 network. This method accelerates the training and improves the final performance. The structure of their model is displayed in Figure 3.2.. For experiments, the DIV2K dataset is used [18].

Haris *et al.* [11] proposed Deep Back-Projection Networks (DBPN). In their network structure, a new way of reducing errors was proposed. In this mechanism, layers with up and downsampling are used, and this process iteratively works across the network. This iterative process is shown in Figure 3.3..

Finally, as shown in Figure 3.4., deep SR Networks can be primarily divided into four types:

- **SR algorithms with predefined up-sampling** The most important thing about this method is the use of a medium resolution (MR) as a standard for the images. For this purpose, some image processing algorithms and numerical analysis such as interpolation (especially bicubic) are used to map the images in the first phase from MR to HR. The pioneering work about this method is [12]. In this work, the researcher tries to use a simple CNN to map images from MR to HR. With the development of CNNs and

their associated parameters, this kind of method also benefits from these developments [13–15].

- **SR algorithms with single up-sampling** In this type of method, unlike the previous method, there are no predefined processes on LR input images. For this purpose, some CNNs have been used that try to transmit the LR image features and construct HR images at the end of the network. The disadvantage of this method is the training time because the CNNs used in this method have a large number of parameters and filters [10, 16–18].

- **SR algorithms with progressive up-sampling** The CNNs show impressive effects in creating HR images in the previous method. Therefore, in this kind of SR algorithm for constructing HR images, a cascade of CNNs is used, which is named the Laplacian Pyramid SR network. In each stage of the pyramid network, the images are gradually up-sampled and finally, SR images are created [19].

- **SR algorithms with iterative up and down-sampling** In general, feed-forward architectures that act as one-way mappings only map rich representations of the input space to the output space. Such an approach is not successful in mapping LR images due to the limited features available. To solve this problem, this method uses iterative up-sampling and down-sampling to obtain the best HR features and minimize the error between iterations [11].

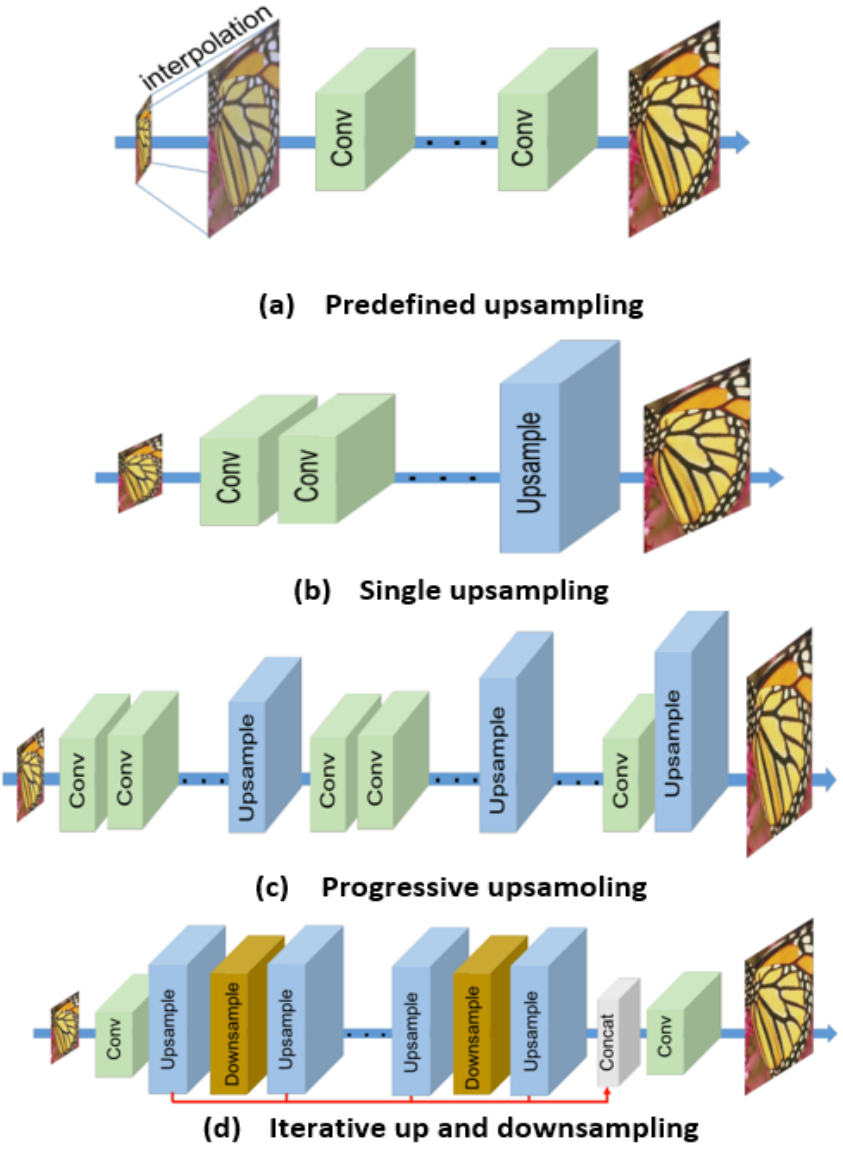

Figure 3.4. Comparison of SR deep networks (a) Predefined up-sampling [12–15], (b) Single up-sampling [10, 16–18] , (c) Progressive up-sampling [19], and (d) Iterative up and down-sampling [11].

4. METHODOLOGY

Most real-world surveillance cases demand recognition of personal attributes from a distant, which convinces the need to process these attributes in low resolution. Our target is to get better person attribute recognition performance from LR images using the SR algorithm besides the MAR network, as the overall idea is demonstrated in Figure 4.1. The proposed architecture is based on SR and attributes recognition networks. The architecture and the resources out there provide opportunities for us in the training process, including datasets, existing high-performance models, and training methodologies to benefit. However, there are caveats in competing with the state-of-the-art models; the resulting extensive network increases the complexity of training and preparing the datasets trained end-to-end.

Our framework consists of joint training of two main parts: i) SR framework and ii) attribute recognition framework. The overall pipeline of the proposed method's end-to-end framework is illustrated in Figure. 4.2.. In the following, the details of these two main components are discussed.

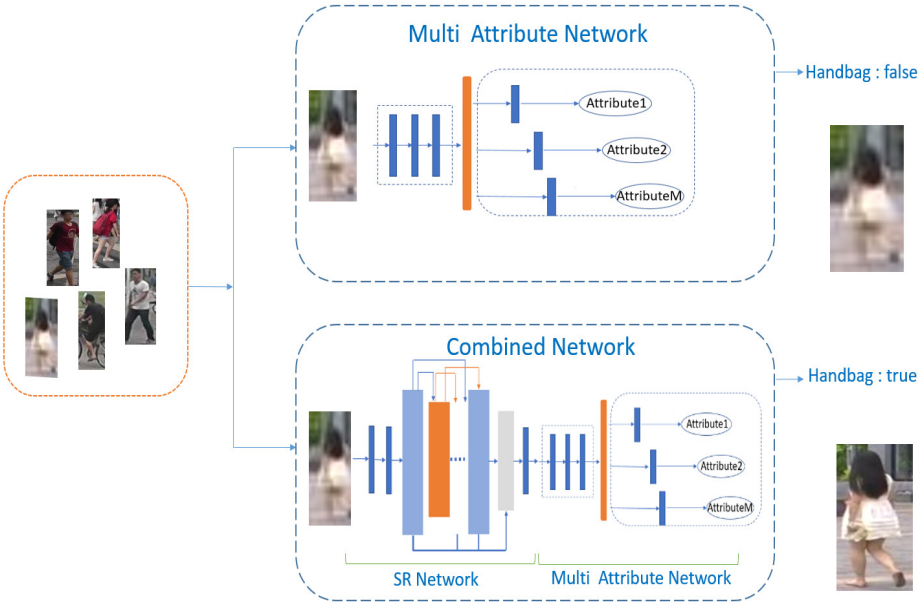

Figure 4.1. Our proposed hybrid network combines the power of a SR network with MAR network to provide better recognition for personal attributes in LR images.

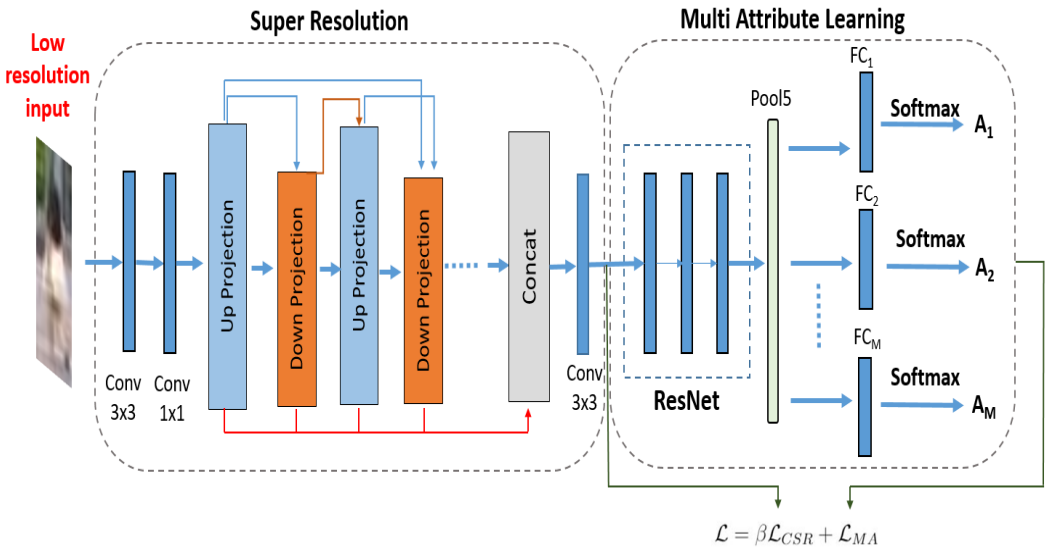

Figure 4.2. The proposed architecture of our SRMAR model.

4.1. Super Resolution Network

The first part of the proposed network is the SR component. Note that we aim to learn a joint network, where the SR components help the recognition of attributes in low resolution. Therefore, any end-to-end trainable SR model can be integrated into our proposed framework in this context. To this end, we utilize two recent state-of-the-art SR models: Enhanced Deep Super-Resolution network (EDSR) [10], and Deep Back-Projection Networks (DBPN) [11].

With the development of deep CNN, especially residual learning techniques [13] exhibit the improved performance of recent SR models, the EDSR SR model [10] extends existing SR networks like [20, 84] that use residual networks by removing additional modules/layers like batch normalization and ReLU. The comparison of residual blocks is shown in Figure. 4.3. With the elimination of batch normalization layers, this elimination increases the performance considerably, and the required memory is reduced by 40% during training compared to SRResNet [20]. This reduction helps improve the model by expanding the size; as a result, EDSR yields better performance.

The simplest way to enhance the performance of the network model, especially CNN, is to stack many layers or increase the number of filters. However, the training procedure is

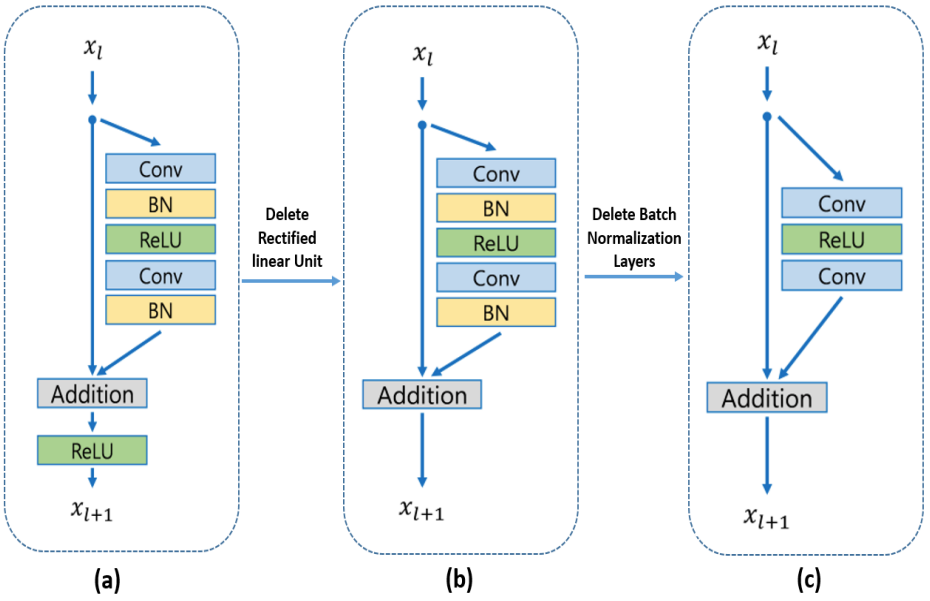

Figure 4.3. The structure of types of residual blocks that are used in (a) ResNet [13], (b) SRResNet [20] , and (c) EDSR [10] .

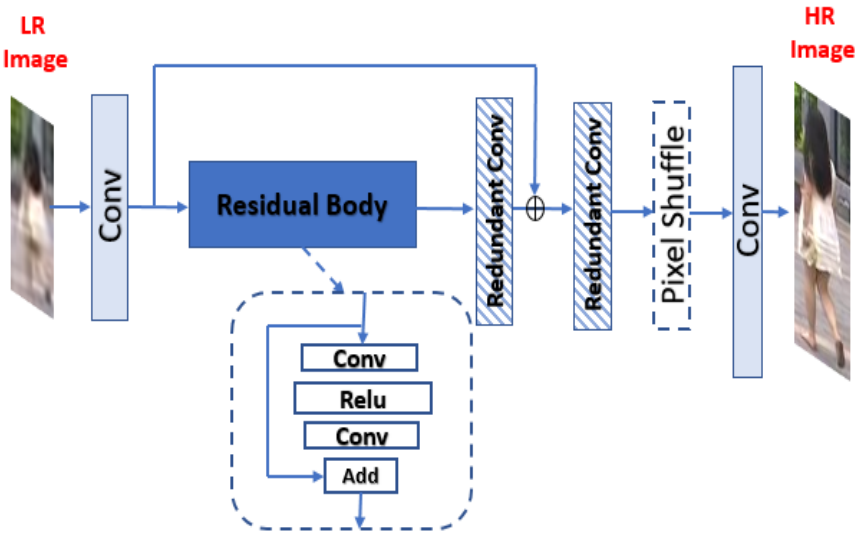

Figure 4.4. Structure of EDSR architecture [10].

directly related to the number of feature maps; after increasing from a certain level, the training process is unstable functionally. To solve this problem in the EDSR network, they adopt residual scaling [85] with a factor equal to 0.1. As a CNN architecture, a residual block is placed after every convolution layer. The structure of this model is shown in Fig. 4.4.

The second SR model that we have used is the DBPN [11] model. DBPN [11] has two stages, the reciprocally connected up and down-sampling stage and the error feedback stage. Generally, feed-forward architectures that act as one-way mapping only map rich representations of the input to output space. Such an approach is not successful in mapping LR images because of the limited features available. To solve this problem, DBPN [11] model generates HR features during up-sampling and, during down-sampling, these features are projected back to LR space. The second stage, the error feedback stage, has a mechanism from the up to down-scaling steps that positively influences the training process to achieve a better reconstruction. The corresponding architecture of the DBPN model [11] is given in Fig.4.5.

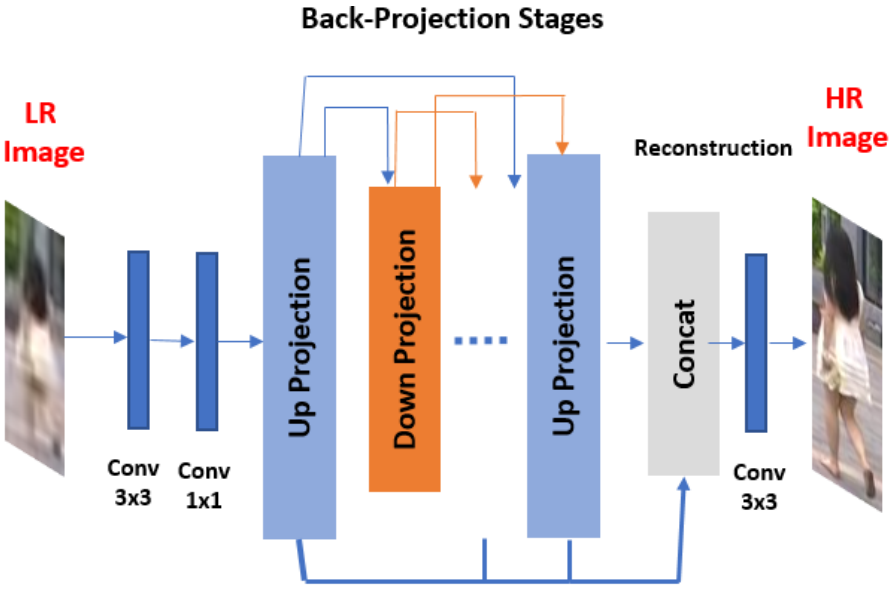

Figure 4.5. Structure of the DBPN model proposed in [11].

Both of these SR networks were trained following to the original papers advises [10, 11] for EDSR and DBPN models respectively, using [86] loss function on the outputs of SR network:

$$\mathcal{L}_{SR}(\hat{I}, I) = \frac{1}{hwc} \sum_{ijk} \sqrt{(\hat{I}_{ijk} - I_{ijk})^2 + \epsilon^2} \tag{1}$$

where h, w, c denote height, width and channels respectively. $\hat{I}$ is network output, I is super resolution ground truth, and $i, j, k \in \{1, 2, 3, \ldots, M\}$ represent the coordinates of the target tensor with dimension. ϵ is a constant for numerical stability, which is usually taken as 0.001.

We combine the $\mathcal{L}_{SR}$ with dice loss function:

$$\mathcal{DC}(\hat{I}, I) = 1 - \frac{2\sum_{ijk} \hat{I}_{i,j,k} I_{i,j,k}}{\sum_{i,j,k} \hat{I}_{i,j,k} + \sum I_{i,j,k}}$$

and get the following loss function:

$$\mathcal{L}_{CSR}(\hat{I}, I) = \mathcal{DC}(\hat{I}, I) + \mathcal{L}_{SR}(\hat{I}, I) \tag{2}$$

which is then used to train the SRMAR model.

4.2. Attribute Network

The second part of the proposed model is a network for learning attributes. Low-resolution images that are upscaled by the SR component are fed into the attribute recognition network for predicting the corresponding attributes. For the MAR task, we adopt the recent network proposed by Lin *et al*. [6]. This network aims for person re-identification and pedestrian attribute recognition at the same time. We adopt the person attribute recognition part of their model, which is trained just on the attribute data set using ResNet-50 [63] as the backbone (Figure 4.6.). This backbone is followed by the attribute recognition, which includes M (number of attributes) Fully Connected (FC) layers followed by a softmax layer [6]. The binary cross-entropy:

$$\mathcal{L}_{MA}(\hat{Y}, Y) = -\sum_{i} Y_i \log(\hat{Y}_i) + \frac{\gamma}{2} \sum_{j} ||w_j||_2 \tag{3}$$

is used as the loss function in training where $\hat{Y}$ is the predicted output and Y is the ground truth label, γ is regularization factor set to 0.02, and w_j represents the weights in the convolution layer j.

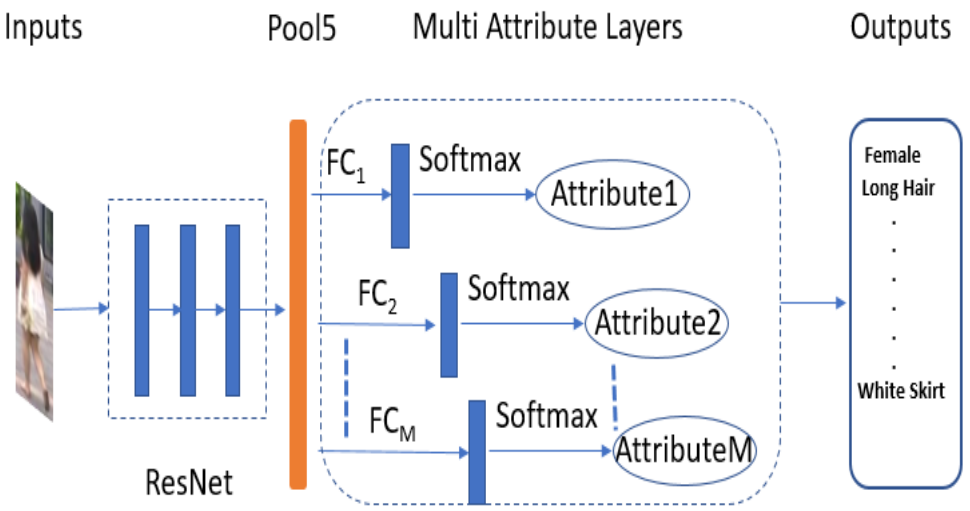

Figure 4.6. Structure of the multi attribute model proposed in [6].

4.3. SRMAR network

For the combined SRMAR neural network, the training process can be handled in different ways. Designing loss functions and optimizers are essential. We have experimented with different loss functions, data augmentation, and other hyper-parameter optimization techniques. Our idea is to design weighted loss based on SR and MAR network loss functions. This helped to reduce the overfitting effect during the training, however, it requires a subtle selection of the regulation parameter β;

$$\mathcal{L} = \mathcal{L}_{MA} + \beta \mathcal{L}_{CSR} \tag{4}$$

Based on results during the training we decided to set $\beta = 0.00005$ which is a trade off between the accuracy and over-fitting of the network. Merging SR and MAR networks leads to a large network that should be trained with care.

4.4. Linear Combination of Models

We also investigate the effect of combining the SRMAR network with the MAR network. The overall architecture for this combination scheme is demonstrated in Fig.4.7. The idea

is to combine the predictive power of the SRMAR model and the MAR model, since our preliminary experiments indicate that some attributes are predicted better by the SRMAR network, whereas some others are predicted by MAR network.

The simplest approach to achieve this is defining equal weights (i.e. $w_k = 1/k,\ k = 2, 3, \dots$) for combining models, which is simple averaging. Here we are trying to use network output statistics on the in-sample data to maximize the accuracy of the linear combination. Obviously one can put other accuracy measures in calculations, but that restricts the problem and makes it difficult to handle with linear optimization techniques. Now the problem is reduced to a linear programming problem. This type of modeling falls in the subject so called data envelopment analysis which appears in different areas of science addressing similar problems [87].

More formally, let $\mu_{i,p}$ and $\mu_{i,n}$ for $i = 1, 2$ be the mean of the outputs of the SRMAR network (i=1) and MAR network (i=2) on positive and negative ground truth data respectively. $\sigma_{i,p}, \sigma_{i,n}, i = 1, 2$ represents the corresponding standard deviations for SRMAR and MA networks. Let ψ_1 and ψ_2 be the outputs of SRMAR and MAR networks respectively having values $[-1,\ 1]$, we want to get weighted average of the two as an output; such that

$$y_{LC} = \mathrm{softmax}(\mathrm{w}_1\psi_1 + \mathrm{w}_2\psi_2) \tag{5}$$

The loss function for the new combined mode can be written as

$$loss_{LC} = \|y_{LC} - y_{tg}\|^2, \tag{6}$$

where y_{tg} is the target output from annotated dataset. The linear combination of the two models is a meta model that takes the pre-softmax output of the sub-models and make decision based on those. Instead of training the combined model, we can use linear optimization to find the optimal weights. The optimization problem that we now should solve to get optimal

weights is formulated as:

$$\begin{aligned} \min_{w_1,w_2} \quad & \|y_{LC} - y_{tg}\|^2, \\ \text{s.t} \quad & y_{LC} = \text{softmax}(\text{w}_1\psi_1 + \text{w}_2\psi_2), \end{aligned}$$

Practically we can calculate w_1 and w_2 based on statistics of the outputs of the two networks so that $y > 0$ for having the attribute and $y < 0$ otherwise. Specially we want w_1 and w_2 to hold the following constraints creating feasible region. If we denote networks output by ψ_i^p and ψ_i^n to indicate that the input possesses the attribute and does not respectively, then the direct reformulation of the last two inequalities will be:

$$\begin{aligned} \psi_1^p w_1 + \psi_2^p w_2 \geq 0, \\ \psi_1^n w_1 + \psi_2^n w_2 < 0, \end{aligned}$$

while maximizing the number of correct predictions. We can calculate the confidence interval of the normalized coefficients as follows: $\psi_1^p \in (\mu_{1,p} - \epsilon_1, \mu_{1,p} + \epsilon_1)$, $\psi_2^p \in (\mu_{2,p} - \epsilon_2, \mu_{2,p} + \epsilon_2)$, $\psi_1^n \in (\mu_{1,n} - \epsilon_1, \mu_{1,n} + \epsilon_1)$ and $\psi_2^n \in (\mu_{2,n} - \epsilon_2, \mu_{2,n} + \epsilon_2)$. Here $\epsilon_i = \mathcal{Z}_{0,1}^i \frac{\sigma}{\sqrt{N}}$. $\mathcal{Z}_{0,1}^i$ for $i = 1, 2$ is the normalization of outputs and $\mathcal{Z}_{0,1}$ stands for normal distribution with mean 0 and variance 1 and N is number of samples (here number of in-sample data). σ stands for the corresponding distribution standard deviation.

In our case we set $W_i = \{\ 0.01 \times j,\ j = 1, 2, \ldots, 100\ \}$ for $i = 1, 2$ and therefore get finite feasible region $W_1 \times W_2$ by which we calculate the target function by selecting the pair $(w_1,\ w_2)$ that lead to the highest accuracy. As the size of training set is large, to solve the resulted integer programming we use the python wrapper of the integer programming solver SCIP [88]. The proposed architecture of this linear combination strategy is demonstrated in Figure.4.7.

Note that the proposed method will perform at least as better as simple average method because the solution $w_1 = 1/2,\ w_2 = 1/2$ is already inside the feasible region of the linear programming model. As demonstrated in the experiments section, this combination strategy

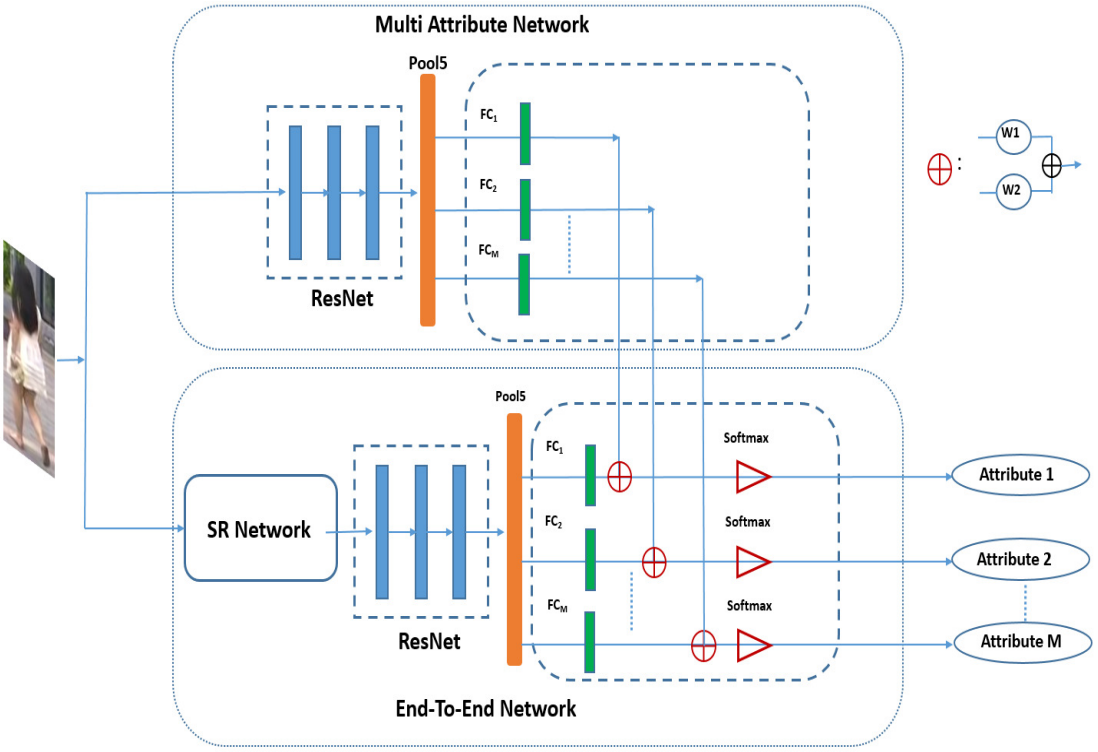

Figure 4.7. Architecture of the proposed linear combination strategy.

have led to better results compared to using a single model. In Appendix B plots of all weights of both models are given.

5. EXPERIMENTS

5.1. Datasets

We carry out our experimental evaluation on two widely used benchmark datasets Market-1501 [1], and DukeMTMC-reID [2].

5.2. Market-1501

Market-1501 [1] is one of the biggest datasets prepared for person re-ID research, containing 32,668 and 3,368 query images, as shown in Figure 5.1. 751 identities (19,732) are used for training and 750 identities (13,328 images) are for testing [6]. There are 27 attributes as shown in Table 5.1. Following [6], we work over 11 attributes: gender (man, woman), hair status (hair.l), sleeve status (slv.l), wearing hat (w.hat), carrying backpack (b.pack) or handbag (h.bag), upper-body cloth color[2] -8 colors- (co.up), lower-body clothing color[3] -9 colors-(co.low), age, lower-body clothing length status (ll.clth) and lower-body clothing type (tl.clth) In this dataset, background and junk images are not considered during training or testing since they do not have the corresponding attribute labels.

5.3. DukeMTMC-reID

The second dataset is DukeMTMC-reID dataset [2] as shown in Figure 5.2.. It contains 702 identities (16,522 images) for training and 702 identities (19,889 images) for testing. 10 attributes are covered in this dataset: gender (man, woman), type of shoe (boots), wearing hat (w.hat), carrying backpack (b.pack), handbag(h.bag), or bag (bag), color of shoes(co.shoes), upper-body clothing length (l.up), upper-body clothing color -8 colors- (co.up) and lower-body clothing color -7 colors-(co.low). The color selections are the same for both up and down clothing colors[4]. All attributes are shown in Table 5.2.

[2]black, white, red, purple, yellow, gray, blue, green
[3]black,pink, white, yellow, purple, gray,green,blue, brown
[4]black, white, red, gray, blue, green, brown

Figure 5.1. Some samples of Market-1501 [1] dataset.

Figure 5.2. Some samples of DukeMTMC-reID [2] dataset.

5.4. Implementation

In the case of multi-attribute recognition, the feature extraction part follows M small sub-nets, each constructed by a convolutional layer, a pooling layer followed by a fully-connected

Table 5.1. Attribute of Market-1501 dataset [1]

attribute	representation in results	label
gender	gender	man(1), woman(2)
hair status	hair.l	short(1),long(2)
sleeve status	slv.l	long(1), short(2)
lower-body clothing length status	ll.clth	long(1), short(2)
lower-body clothing type	tl,clth	dress(1), pants(2)
hat	w.hat	false(1), true(2)
having backpack	b.pack	false(1), true(2)
having bag	bag	false(1), true(2)
having handbag	h.bag	false(1), true(2)
age	age	young(1), teenager(2), adult(3), old(4)
upper-body clothing color	co.up	false(1), true(2)
lower-body clothing color	co.low	false(1), true(2)

Table 5.2. Attribute of DukeMMTC-erID dataset [2]

attribute	representation in results	label
gender	gender	man(1), woman(2)
upper-body clothing length	l.up	short(1), long(2)
wearing boots	boots	false(1), true(2)
wearing hat	w.hat	false(1), true(2)
having backpack	b.pack	false(1), true(2)
having bag	bag	false(1), true(2)
having handbag	h.bag	false(1), true(2)
shoes color	c.shoes	dark(1), light(2)
upper-body clothing colors	co.up	false(1), true(2)
lower-body clothing colors	co.low	false(1), true(2)

(FC) layer, and finally, the softmax function. We use the original backbone network (ResNet-50)[63], which was pre-trained on ImageNet. In the training stage of the MAR network, we use the loss function as defined in Eq. 3. SR network which has a kind of encode-decoder architecture, Eq.1 is considered as loss function. In training SR and MAR, Adam optimizer is used. When training the SRMAR network which is based on SR and MAR networks subsequently, we fix the SR module except for the last 15 layers. The plots of our model's

loss and accuracy on both train and validation sets are given in Appendix A as can be seen in the Figures, there are three phases in the results; first a rapid improvement (sharp accuracy increase) at the second phase there is a slow improvement in the accuracy on test data and in the third phase, we see improvement in the train set accuracy and slow decrease in test accuracy. Model checkpoints are taken at the end of phase two.

Our preliminary experiments indicate that freezing more layers leads to poor results while freezing fewer layers causes over-fitting. During training SRMAR network, DiffGrad optimizer [89] with a cyclic learning schedule is used to optimize the combined loss function (Equation 4). The maximum size of images in selected datasets is 64×128; therefore, we consider this as the reference size. Paying attention to the input image sizes of SR models EDSR (2x, 3x, and 4x) and DBPN (2x, 4x, and 8x), we downsize images with bi-cubic interpolation into four sizes: 32×64, 21×42, 16×32, and 8×16 respect to SR model input sizes. For training, we set the batch size to 32 in all experiments, with an initial learning rate set to 0.001 with a learning scheduler multiplying the learning rate by 0.1 every five epochs. Moreover, the networks are trained for 40-60 epochs. Sample input images are shown in Figure 5.3. and predictions for the sample inputs are presented in Table 5.3.

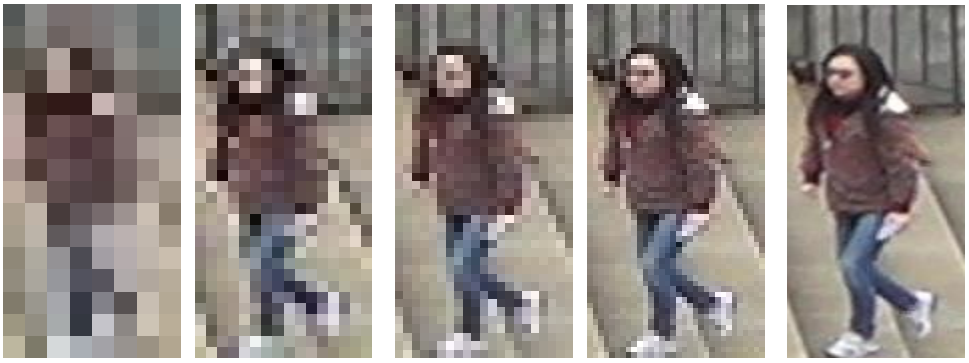

Figure 5.3. Sample image from DukeMMTC-reID: from left to right; base image size 8x16, 16x32, 21x42, 32x64 and original image. For visualization, all images is visualized in a fixed resolution so that small images were interpolated and larger images were down-sampled.

Table 5.3. Ground truth and inference results for the input images of Figure 5.3.

attribute	ground truth	SRMAR-E prediction		
		16x32	21x32	32x64
gender	1	1	1	1
w.hat	0	0	0	0
boots	0	0	0	0
l.up	1	1	1	1
b.pack	1	0	1	1
h.bag	0	1	0	0
bag	0	0	0	0
co.shoes	1	1	1	1
co.up	7	7	7	7
co.low	4	4	4	4

5.5. Experimental Results

We evaluate our SRMAR model via extensive experiments over the two benchmark datasets. There are two versions, SRMAR-E is uses EDSR [10] SR model within the joint network, whereas SRMAR-D uses DBPN [11] SR model. We compare the proposed SRMAR model with the MAR[6] model that is applied to the same resolution images. We test several resolutions such as (32×64), (21×42) and $(16 \times 32$ or $8 \times 16)$. The number of total attributes is 30 and 23 for the Market-1501 and the DukeMTMC-reID datasets, respectively. For the sake of representation, we summarized them into 11 (Market-1501) and 10 (DukeMTMC-reID) attributes by averaging similar attributes that belong to one category (such as upper body colors or lower body colors). The rightmost column represents the average accuracy for each method processing over the presented image resolution.

In Table 5.4., the results of the SRMAR-E method over the DukeMTMC-reID dataset is shown. According to Table 5.4., for all the resolutions, the SRMAR-E model improves the recognition performance of the MAR model significantly. For 16×32 resolution input size, the original model without any SR component achieves an accuracy of 68.78%, whereas the

Table 5.4. Accuracy results on Market-1501 dataset using EDSR SR model

model	img size	gender	age	hair.l	slv.l	ll.clth	tl.clth	b.pack	h.bag	bag	co.up	co.low	mean
	16×32	56.12	74.27	63.11	80,34	62.23	53.51	64.05	86.27	65.57	77.63	73.62	68.78
MAR [6]	21×42	54.91	75.00	63.12	**93.5**	67	87.17	72.22	87.17	65.91	78.87	80.05	74.98
	32×64	56.12	75.54	63.93	**93.52**	67.52	88.00	74.91	**90.57**	71.54	85.12	82.57	77.21
	16×32	69.41	93.83	74.65	**93.53**	78.65	88.41	76.12	**90.40**	70.93	90.81	88.71	83.22
SRMAR-E	21×42	72.12	92.1	74.94	93.43	77.48	87.81	**75.32**	**90.26**	62.23	89.61	88.31	82.14
	32×64	60.91	**94.1**	62.01	93.51	**78.76**	87.14	**75.39**	90.51	69.51	90.71	87.52	80.91
	16×32	**69.44**	**94.37**	**76.00**	93.35	**80.00**	**88.41**	**79.42**	90.14	**76.32**	**90.91**	**89.72**	**84.37**
Combined-E	21×42	**79.43**	**94.22**	**83.61**	93.42	**80.57**	**89.31**	75.22	90.24	**74.21**	**89.77**	**91.43**	**85.58**
	32×64	**71.32**	93.61	**71.93**	93.51	71.64	**88.51**	73.77	90.51	**73.76**	**91.74**	**91.44**	**82.89**

Table 5.5. Accuracy results on Market-1501 dataset using DBPN SR network.

model	img size	gender	age	hair.l	slv.l	ll.clth	tl.clth	b.pack	h.bag	bag	co.up	co.low	mean
	8×16	54.20	71.2	62.52	75.91	62.12	53.01	47.80	73.7	64.06	76.31	78.61	65.45
MAR[6]	16×32	56.12	74.27	63.11	80,34	62.23	53.51	64.05	86.27	65.57	77.63	73.62	68.78
	32×64	56.12	75.54	63.93	93.52	**67.52**	88.00	74.91	**90.57**	71.54	85.12	82.57	77.21
	8×16	54.50	**94.12**	63.45	93.61	70.91	**89.11**	74.44	90.21	75.57	88.36	87.51	80.16
SRMAR-D	16×32	55.32	**93.60**	63.90	93.50	68.24	**88.10**	74.90	**90.50**	**75.70**	88.60	86.70	79.91
	32×64	55.91	93.7	62.92	**93.73**	66.91	**88.37**	74.96	90.54	**75.76**	88.63	85.41	79.71
	8×16	**55.05**	93.61	**63.92**	**93.84**	**71.01**	88.12	**74.87**	**90.52**	**75.71**	**88.63**	**88.47**	**80.34**
Combined-D	16×32	**55.91**	93.22	**63.94**	**93.51**	**68.85**	87.92	**75.00**	90.23	75.32	**88.92**	**88.21**	**80.09**
	32×64	**58.44**	**93.71**	**64.15**	93.37	67.12	87.91	**77.35**	90.32	75.41	**90.54**	**90.72**	**80.82**

proposed SRMAR-E model achieves 83.22% accuracy. The accuracy is even more improved to 84.37% when the MAR network is combined with the SRMAR network using the proposed linear combination strategy (Combined-E). Similarly, for the input size 21×42, the proposed SRMAR-E model improves the accuracy of the MAR [6] model from 74.98% to 82.14%, and the linear combination of the two models (Combined-D) achieve an accuracy of 85.58%. For the 32x64 input size, even though the improvements are not that drastic, still the accuracies improve from 77.21 % to 82.89%.

Table 5.5. presents the similar experiments using the SRMAR-D (that utilizes DBPN model as the SR component). We observe that the SRMAR model improves over MAR [6] from 77.21% to 79.71% for large input, from 74.98% to 79.91% for medium input, and from 65.45% to 80,16% for small input. Like the SRMAR-E, we observe that the relative improvement in accuracies for small size input (8×16) is more than the other two input sizes.

Table 5.6. Accuracy results on DukeMTMC-reID dataset using EDSR SR model

model	img size	gender	w.hat	boots	l.up	b.pack	h.bag	bag	co.shoes	co.up	co.low	mean
	16×32	53.33	57.57	76.24	84.32	51.90	75.78	**82.91**	78.21	80.14	76.75	71.71
MAR [6]	21×42	55.15	68.17	66.40	87.84	54.3	93.62	82.37	87.90	90.8	83.02	77.05
	32×64	58.81	75.01	77.41	87.74	54.86	92.84	**83.53**	88.00	90.51	82.27	79.09
	16×32	69.22	**77.37**	77.21	84.67	67.22	92.67	81.54	**86.51**	92.30	87.03	81.57
SRMAR-E	21×42	63.24	78.02	78.81	87.93	67.94	93.66	82.73	87.84	92.13	87.14	81.94
	32×64	74.42	79.51	80.92	86.34	**72.21**	92.15	80.64	85.63	91.17	87.52	83.05
	16×32	**73.5**	76.94	**81.12**	**86.23**	**69.44**	**93.30**	81.64	86.23	**92.61**	**88.94**	**82.99**
Combined-E	21×42	**69.61**	**78.45**	**78.94**	**88.12**	**69.57**	**93.88**	**83.64**	**88.91**	**93.22**	**87.41**	**83.17**
	32×64	**77.93**	**79.81**	**82.87**	**88.43**	71.01	**93.66**	82.83	**88.07**	**92.85**	**87.74**	**84.52**

Table 5.7. Accuracy results on DukeMTMC-reID dataset using DBPN SR model

model	img size	gender	w.hat	boots	l.up	b.pack	h.bag	bag	co.shoes	co.up	co.low	mean
	8×16	42.37	55.31	76.14	75.90	54.91	73.48	79.21	66.27	80.07	76.54	68.02
MAR [6]	16×32	53.33	57.57	76.24	84.32	51.90	75.78	82.91	78.21	80.14	76.75	71.71
	32×64	58.81	75.01	77.41	87.74	**54.86**	92.84	83.53	88.00	90.51	82.27	79.09
	8×16	58.62	74.73	77.41	**87.88**	54.63	93.66	83.62	**88.14**	87.72	83.39	78.98
SRMAR-D	16×32	61.44	74.91	77.52	87.73	54.74	93.71	83.54	88.47	87.82	84.14	79.40
	32×64	61.72	75.15	**77.91**	87.54	53.66	**93.87**	83.91	**87.82**	87.43	84.44	79.34
	8×16	**58.75**	**75.22**	**77.82**	87.87	**74.65**	**93.68**	**83.62**	88.11	**90.46**	**84.31**	**81.44**
Combined-D	16×32	**62.85**	**75.53**	**78.84**	**88.00**	**71.62**	**93.76**	**83.75**	**88.49**	**92.81**	**89.61**	**82.52**
	32×64	**63.31**	**75.73**	77.71	**87.87**	54.83	93.80	**83.91**	87.81	**91.58**	**85.92**	**80.24**

Also, we observe that the linear combination of two models is again effective in increasing the overall accuracies even further.

For the DukeMTMC-reID dataset, we observe similar trend in our experiments. Table 5.6. and Table 5.7. presents the corresponding results. In Table 5.6., the results of the SRMAR-E model is presented. The overall accuracies are increased from 79.09% to 83.05% for 32×64 sized input, from 77.05% to 81.94% for 21×42 sized input, and from 71.71% to 81.57% for input sizes 16×32. Again, the improvements are quite significant, especially when used with lower resolution images. As shown in Table 5.6., the proposed linear combination of MAR [6] and SRMAR-E models are also effective, offering a performance improvement more than 1% over SRMAR-E model.

Table 5.7. shows the performance of the proposed SRMAR-D model, and the following

improvements are achieved over the reference MAR model[6]: for 32×64 size input, the overall accuracy has increased from 79.09% to 79.34%, for 16×32 sized input, the accuracy has increased from 77.05% to 79.40%, and for the 8×16 input from 68.02% to 78.98%. We observe the same pattern in the value of improvements w.r.t sizes; the smaller size, the better improvement.

From the results in Table 5.4.-5.7., we can further evaluate the performance of individual attributes. In the case of the Market-1501 dataset, from Table 5.4., the attribute that has the lowest recognition rate is the gender attribute, whereas the long sleeve (l.slv) attribute seems to be the attribute with the highest recognition accuracy for the MAR model[6] in different resolutions. SRMAR-E model improves the recognition accuracy of gender attribute significantly. The performance improvement is also remarkable for the age, style of clothing (s.clth), color of up clothing (c.up) and color of down clothing (c.down) attributes. For some of the resolutions, especially for 21×42 and 32×64 input sizes, for "bag" and "hair" attributes, there is a reduction in accuracy in Table 5.4.. In such cases, using the linear combination of these models as proposed helps. As can be seen, the linear combination strategy resolves accuracy reduction of the SRMAR-E for "bag" attribute from 69.51% to 73.76% for 32×64 sized input, from 62.23% to 74.21% for 21×42 sized input, and from 70.93% to 76.32% for 16×32 sized input.

Table 5.8. Comparison of overall accuracy of models using images of 16×32 size as input.

Model	Market1501	DukeMTMC
MAR [6]	68.78	71.71
SRMAR-D	79.91	79.40
SRMAR-E	83.22	81.57
Combined-D	80.09	82.52
Combined-E	**84.37**	**82.99**

Table 5.8. summaries the experimental results for 16×32 resolution input size over both datasets. As can be seen from this table, SRMAR-E model that uses the EDSR [10] SR model performs better than the SRMAR-D model that uses DBPN [11] SR model on both of the benchmark datasets. Moreover, the proposed linear combination or SRMAR and MAR

models offer a notable increase in the accuracy of both SRMAR-(E, D). We can say that combining both SRMAR and MAR models in the proposed way offers the best recognition performances for the recognition of person attributes in low resolution images.

In the following we will calculate relative improvement with the following formula:

$$\text{relative improvement} = \frac{Combined_{MA} - Reference_{MA}}{Reference_{MA}} \tag{7}$$

where, $Combined_{MA}$ stands for Combined model and $Reference_{MA}$ for reference attribute model mean average accuracy on test dataset.

According to Table 5.4., for all the resolutions, we observe 0.0479 for 32×64 sized input, 0.0955 for 21×42 sized input, and 0.210 for 16×32 sized input improvement in average accuracies. According to Table 5.5., for all the resolution, we observe 0.0323, 0.0617, and 0.224 improvements in average accuracies concerning 32×64, 21×42, and 8×16 input sizes. Like these two tables, with pay attention to Table 5.6. and Table 5.7., we observer 0.050 for 32×64, 0.0635 for 21×42 and 0.1375 for 16×32 input sizes improvement and see 0.0032 for 32×64, 0.0305 for 21×42 , and 0.1610 for 8×16 input sizes improvement in average accuracies respectively to Tables.

The advantage of the linear combination model will be clear if we consider mean average improvement with respect to the SR and reference models; results are shown in Figures 5.4. to 5.7.. For the Market-1501 dataset and in the case of the SRMAR-E model; For large input, we see 0.0245 improvement relative to the SRMAR-E and 0.0736 improvement relative to reference model. In medium input, there is a 0.0419 improvement relative to the SRMAR-E and 0.1414 improvements relative to the reference model. For small input, we get 0.0138 improvements relative to the SRMAR-E and 0.2267 improvements relative to reference model. The highest average improvement is obtained for small sizes and with respect to the reference model. The full comparison of improvements obtained by linear combination for the Market-1501 and the DukeMTMC-reID datasets are represented in Figures 5.4. to 5.7.

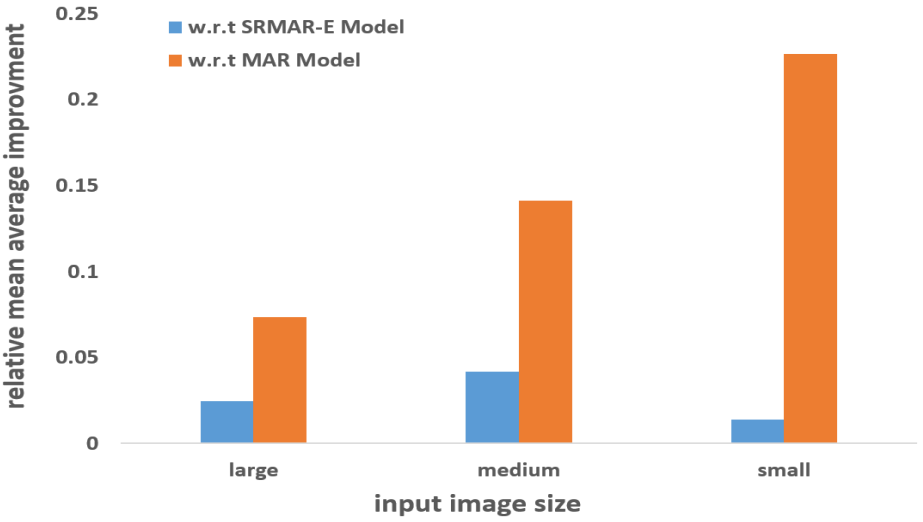

Figure 5.4. Linear combination model improvement relative to SRMAR-E, and MAR model results in Market-1501 in three input sizes large (32×64), medium (21×42), and small (16×32).

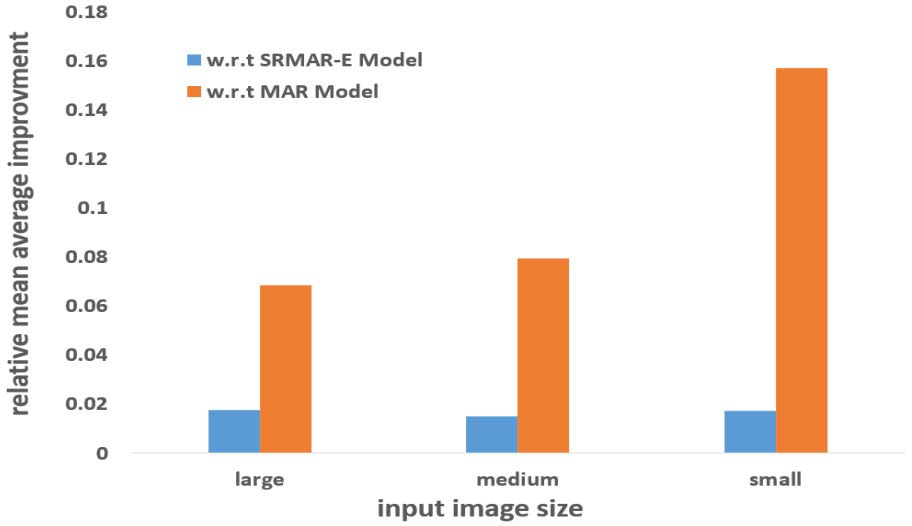

Figure 5.5. Linear combination model improvement relative to SRMAR-E, and MAR model results in DukeMTMC-reID in three input sizes large (32×64), medium (21×42), and small (8×16).

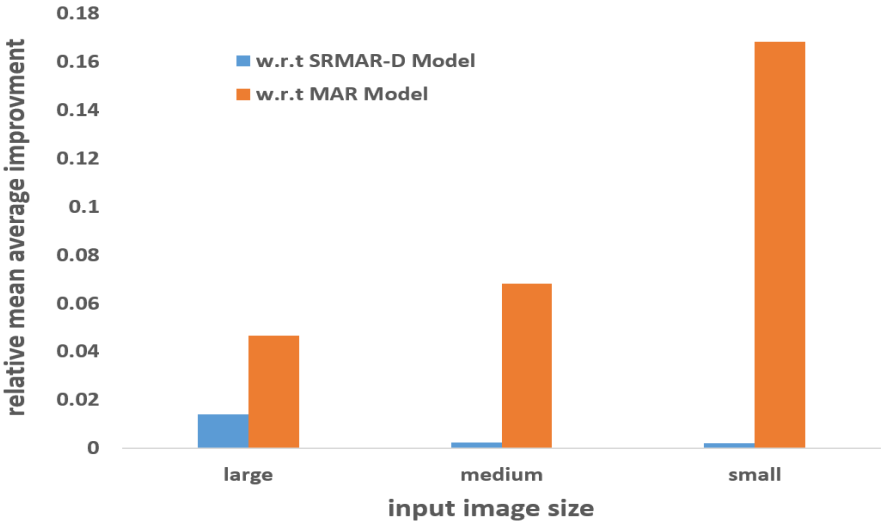

Figure 5.6. Linear combination model improvement relative to SRMAR-D, and MAR model results in Market-1501 in three input sizes large (32×64), medium (21×42), and small (16×32).

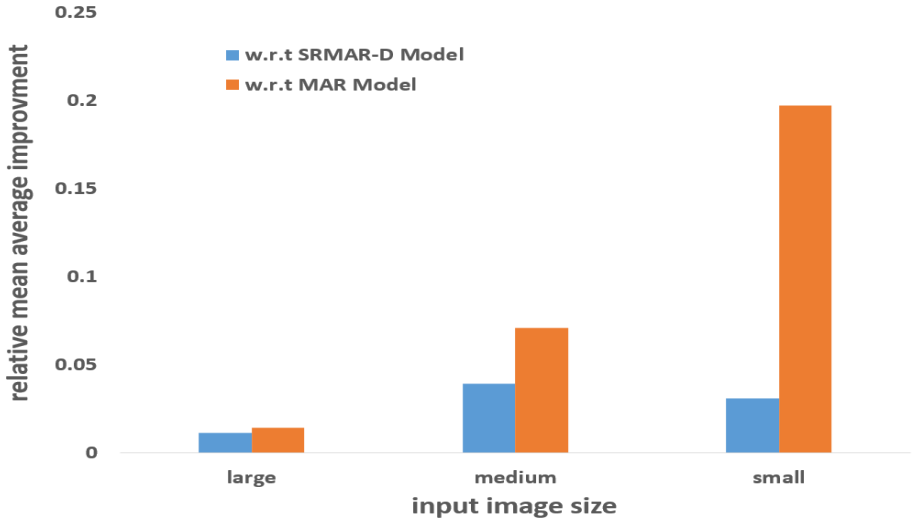

Figure 5.7. Linear combination model improvement relative to SRMAR-D, and MAR model results in DukeMTMC-reID in three input sizes large (32×64), medium (21×42), and small (8×16).

The optimal weights (w_1^*, w_2^*) are calculated using the linear optimization model of section 4.4. per datasets, resolutions, and SR models. As shown in Figure 5.8. and 5.9. which are chosen from the minimum size (input sizes 8×16) of two datasets , the blue part of the columns is the coefficient of the merged models and the orange part is for the multi-attribute model (without SR). As it is visually clear, merged model weights dominated the multi-attribute model weights, meaning that for the optimal linear combination model merged model results are more important than a multi-attribute model, which is another sign of the role of SR in improving model performance. All result's Figures are shown in Appendices B.

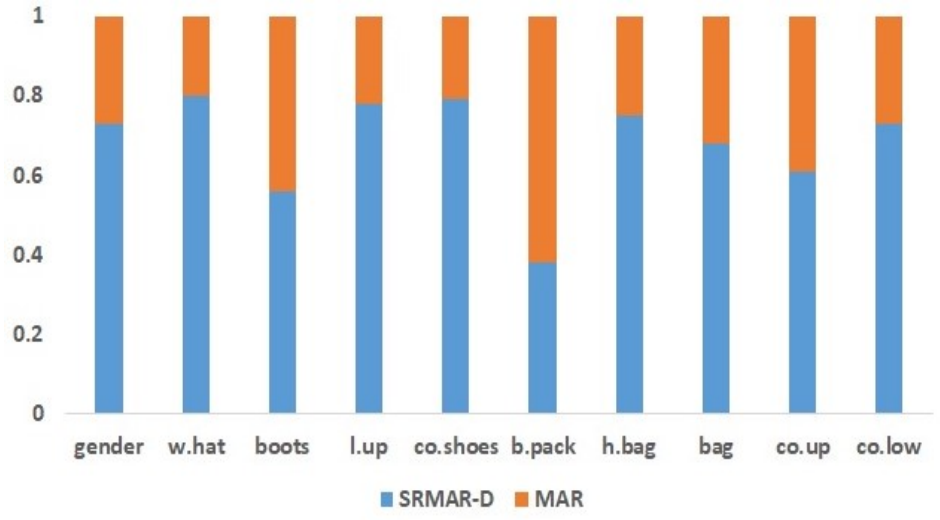

Figure 5.8. The plot of Linear combination model weights concerning SRMAR-D and MAR model getting weights in DukeMTMC-reID. (input sizes 8×16)

5.6. Performance Evaluation of Super Resolution Network

Together with designing an architecture to improve the performance of the multi-attribute model, as already mentioned, as a regulator, the loss function of the SR (Equation (1)) is added to the merged model loss with the coefficient of 0.00005. During these experiments, we also track the behaviour of the re-trained SR model to see whether or not the SR model performance improves by means of joint training.

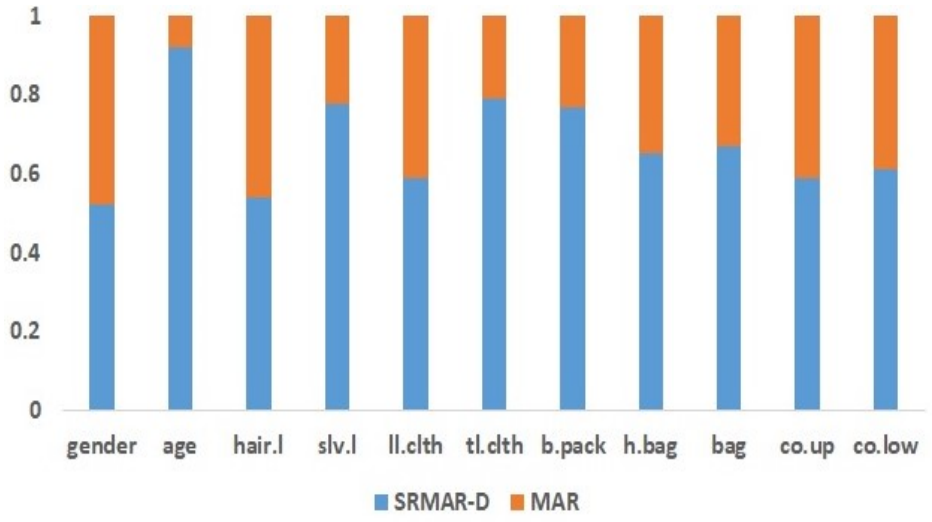

Figure 5.9. The plot of Linear combination model weights concerning SRMAR-D and MAR model getting weights in Market-1501. (input sizes 8×16)

As it is shown in the example SR images in the Figure 5.10., we observe that this way of training does not improve the performance of the SR model. In this experiment, since the output image size by the SR model is larger that the input image, in our case twice as large as the input image, to make loss function work, we resize the SR output using nearest neighbour to the size of the input image.

In this experiment that is conducted to check the SR performance using pre-trained DBPN super-resolution model, Mean Average Error (MAE) on test data is measured. For pre-trained SR model, MAE is measured as 250.17 and for the partially trained (only 15 last layers are trained) SR model we get 5712.67 which shows a dramatic performance loss for SR model. As seen in the Figure 5.10. the performance loss is also visually apparent, since the reconstructed images are of poor quality. In this Figure, the (a) and (c) columns are the outputs of the pre-trained SR model and the (b) and (d) columns are the trained SR model outputs on the corresponding images. Red and blue channel dominance in the pixels reveals weak reconstruction of the given image by the model.

This could be as a result of the effect of the small coefficient that is used in joining the loss functions during training. Since the SR network is trained with a small coefficient, it is likely that the most of the network's effort is put towards optimizing the attribute recognition

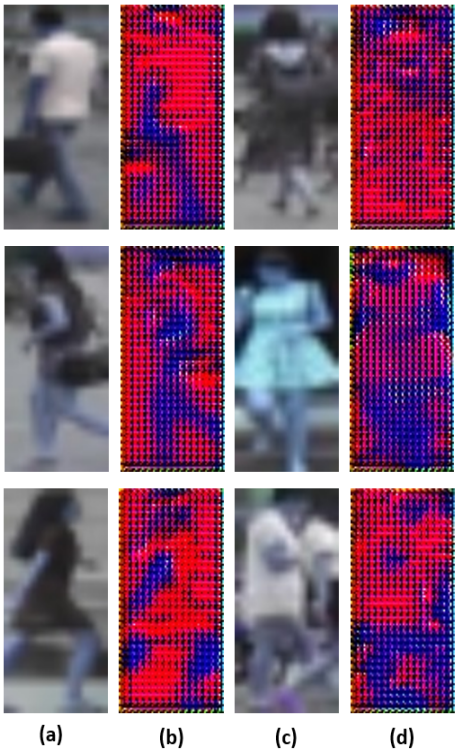

Figure 5.10. Some samples of input images (a, c) and SR output (b, d) of related images.

performance, which harms the SR performance. We think that as there are lots of possibilities to train SR and MA networks at the same time, different weighting, or loss functions can cause improvements on the result of the SR model as well. Although the reconstruction results are not good at all, there is sign of reconstruction in the input images, this means we can improve the results by different hyper-parameters or different training styles which is out of the scope of the current thesis.

6. CONCLUSION AND DISCUSSIONS

Identifying personal attributes in low-resolution images is crucial for surveillance applications; however, the literature rarely addresses this task. Therefore, to fill this gap, in the thesis, we suggested a different approach to utilize the other capacities of the neural networks. We are designing a meta-model in which we take advantage of the different state-of-the-art networks and the meta concept itself.

According to extensive research on the application of neural networks in vision tasks, there is a clear relationship between the quality of input images and the accuracy of the vision task performed by the neural network. This is the primary motivation for the design of the proposed meta-model. However, there is a trade-off between increasing the input image quality and computational cost. Running hundreds of CNNs on many pixels per frame could be highly costly. This means it is not a one-way approach to increase the quality of the images to increase the accuracy of the neural network prediction in multi-attribute recognition. Besides that, there could be technical difficulties in streaming high-quality images, processing them, and storing them. All of those steps could be costly.

In the current thesis, the representation capacity of the neural networks is applied to address the problem mentioned above; the main idea is; that instead, we can store part of the information inside neural networks. The super-resolution network is a feature representation network that we used to interpolate low-quality images and feed them to the multi-attribute recognition network.

This context evaluates the effects of super-resolution CNN architectures on improving multi-attribute recognition performance in low-resolution images. For this purpose, we first control the effects of using some super-resolution network inference on low-resolution images and make new images with the high-resolution dataset. Then, we compare the result of the mean accuracy of initial low-resolution images as input dataset for a multi-attribute network with the result of the mean accuracy of produced high-resolution images used as input dataset to

the same network. The results suggest using high-resolution images improves mean accuracy in the multi-attribute recognition tasks.

After noticing that the accuracy of low-resolution images with using super-resolution increased in separate networks, we plan to make a combined network that started from a super-resolution network and ended with the multi-attribute network as an end-to-end structure. To this end, we adopt one of the state-of-the-art models proposed for multi-attribute recognition and two different super-resolution network architectures and then construct a combined architecture entitled SRMAR-(E, D). To the best of our knowledge, the proposed model of our thesis is the first combined learning model for multi-attribute recognition in low-resolution images. We also propose a linear combination scheme to combine the proposed SRMAR network with the base multi-attribute recognition network. The experiments are carried out in two benchmark datasets and confirm the thesis's claims. Referring to the experimental results presented in Tables 5.4.-5.7., significant improvements are observed in the results as a result of using the super-resolution network. For example, SRMAR-D in the resolution 8×16, mean average precision 68.02 is significantly improved the prediction to 78.98 in the DukeMTMC-reID dataset and 80.16 in the Market-1501 dataset. The same improvement can be seen in the SRMAR-E model too. These processes are repeated multiple times to confirm that they are not by chance. The highest improvement is achieved for the smallest input size. This is proof of the thesis claim. A lot of information is stored inside the super-resolution network, as the theory expected.

The experimental results demonstrate that, for the input images in low resolution, the proposed end-to-end convolutional architecture successfully improves the recognition performance of the base model for person attribute recognition. The second subject that we experiment is combining the features of the models with and without a super-resolution network. These features are 2×1 vector per-attribute for each model. Although the feature space is very small, it led to an improvement in the mean average precision of attribute recognition.

This improvement is achieved by designing a kind of meta-model and training is performed by modelling a linear programming problem to make it very fast. There are lots of different

designs for such a meta-model, however, we select a linear and convex one. This simple meta design gave us an enough improvement to consider it an effective approach. Improvements are significant with respect to MAR (without super-resolution) model, and still is performs better that SRMAR-D/E alone.

6.1. POSSIBLE DIRECTIONS FOR FUTURE WORK

We examined the possibility of improving multi-attribute classification using a super-resolution network during the research. A super-resolution network essentially does to store some information in the network memory so that we can make inferences faster. This is the first point; we can expand this idea to store part of the information inside the network. This idea is already applied in storing a 3D model of an object as a network so it can be generalized to the other areas such as super resolution, image restoration and etc.

The second opinion of the research is the modular design of artificial intelligence agents. For example, we may take a big problem and reduce it to some minor problems and then solve a big problem by putting them together. This helps us to take advantage of state-of-the-art architectures. Moreover, this approach can be generalized to other problems.

Ensambling is a powerful tool in machine learning, but we can create meta-models using the small models to tackle challenging problems. As discussed earlier, meta-model design can be created in a several different ways. The feature spaces can be much bigger that what we used in this paper. Figuring an optimal feature space for the meta-model requires an extensive experiments on the datasets which could be the subject of future researches.

We apply linear combination of two different models that gave us significantly better results. However, the nonlinear combination is also another way to go. There are many problems in which it is required to detect a tiny object in the scene; this approach could be very efficient for these problems, e.g., small magnifying images then feeding to the detector. Scene prediction is the other topic that can be studied in this context. The core idea is to make a network that can understand scene from partially observable information i.e. low resolution images/signals.

As mentioned in 5.6. the possibility of improving the result of SR model, while training merged model, is observed during the experiments. Although, using SR loss as a regularizor does not improved the performance of the SR model, results revealed that visually, the model somehow reconstructed the input image. Starting from here and using different loss functions and hyper-parameters to improve the SR performance at the same time could be a subject for the future work. As a future work, it can be considered to train SR model at the same time with MA model. The merged model loss function can be designed in a different way, with different weights or functions based on the results of the experiments. The core idea is to give SR model a feedback from MA model in a controlled way to improve MA output performance while keeping SR model performing very well as a independent network.

Appendix A

Loss and accuracy plots of training and validation sets

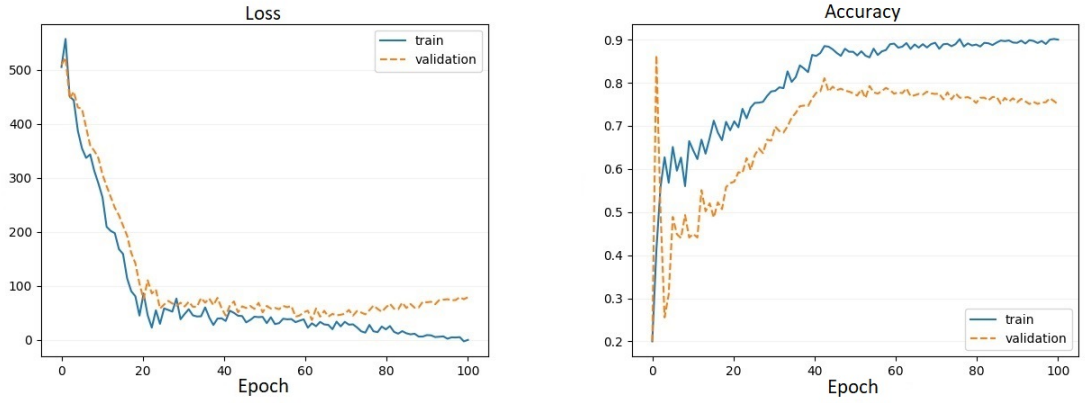

Figure 0.1. The SRMAR-D model training performance plot for the DukeMMTC-reID dataset (input image size is 8x16).

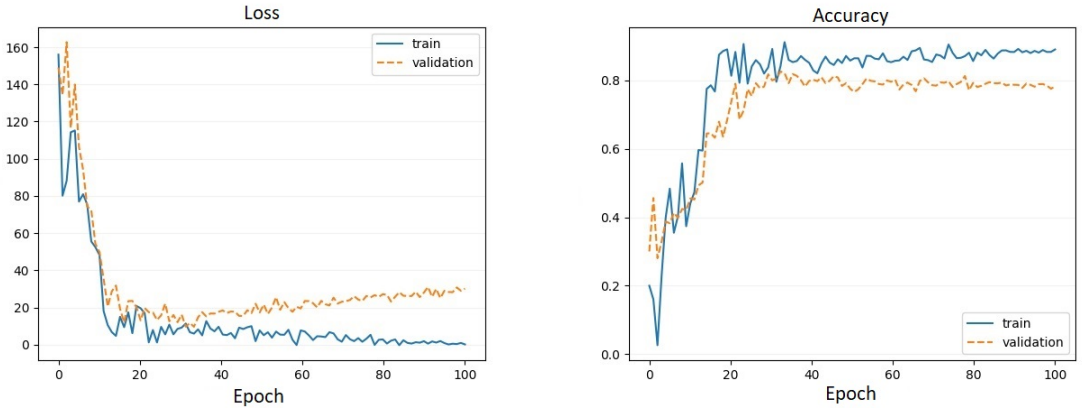

Figure 0.2. The SRMAR-D model training performance plot for the Market-1501 dataset (input image size is 8x16).

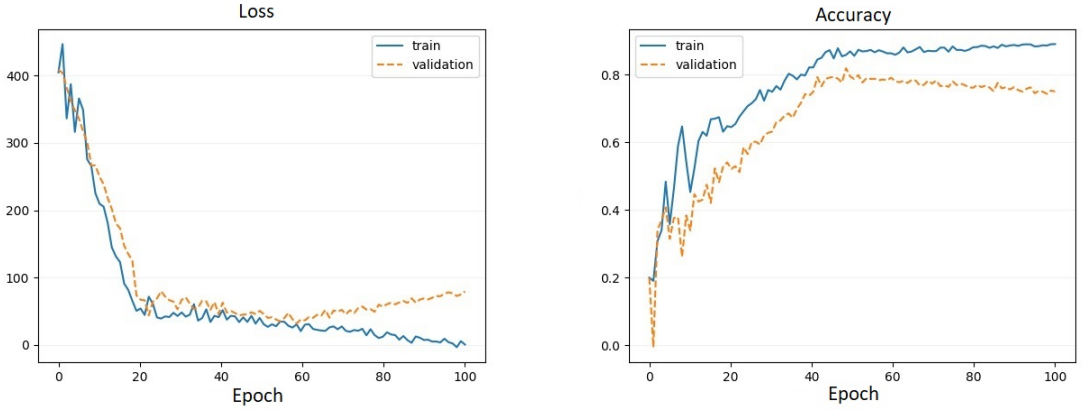

Figure 0.3. The SRMAR-D model training performance plot for the DukeMMTC-reID dataset (input image size is 16x32).

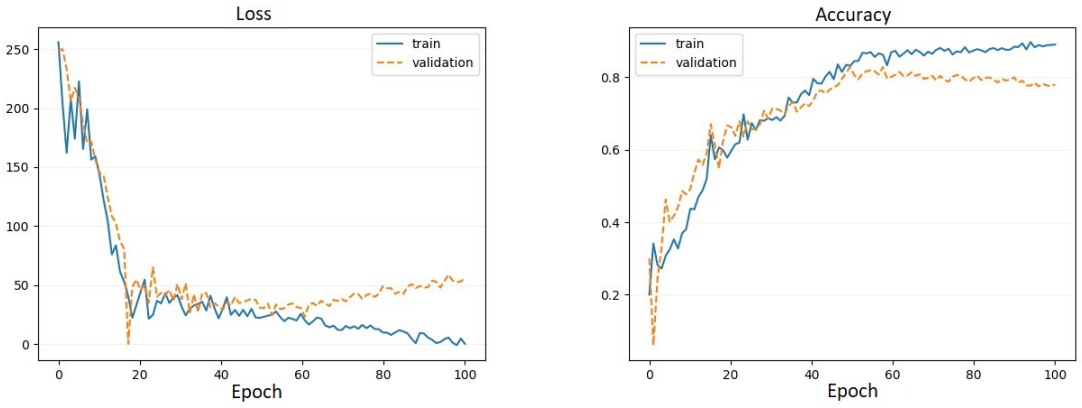

Figure 0.4. The SRMAR-E model training performance plot for the DukeMMTC-reID dataset (input image size is 16x32).

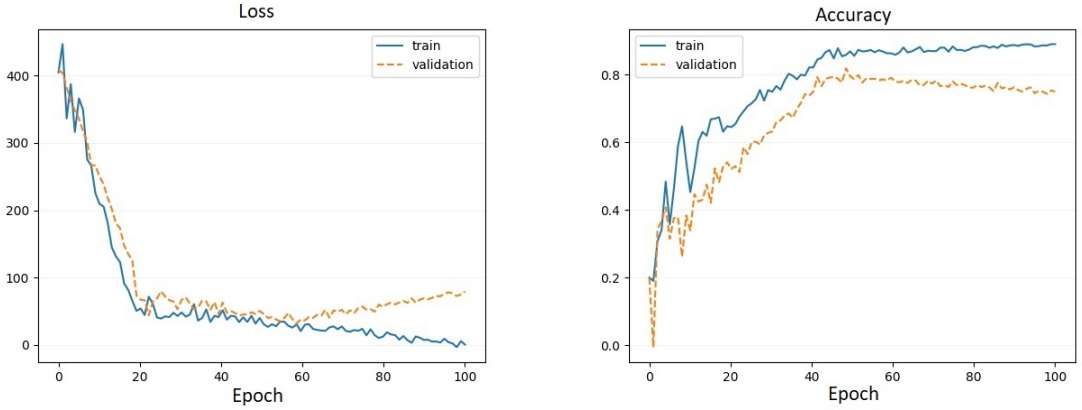

Figure 0.5. The SRMAR-D model training performance plot for the Market-1501 dataset (input image size is 16x32).

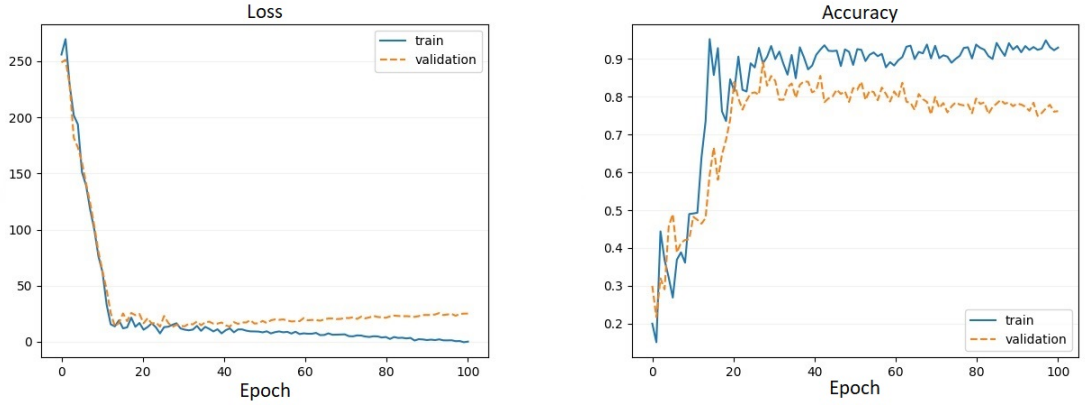

Figure 0.6. The SRMAR-E model training performance plot for the Market-1501 dataset (input image size is 16x32).

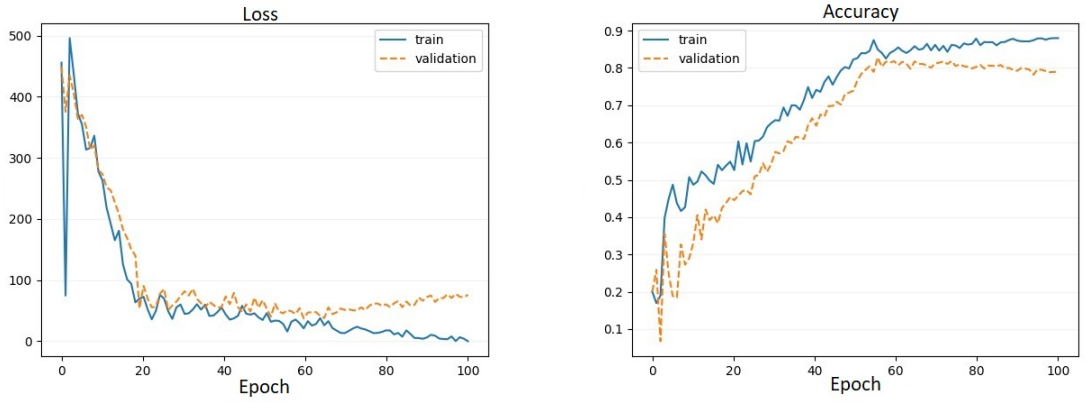

Figure 0.7. The SRMAR-E model training performance plot for the DukeMMTC-reID dataset (input image size is 21x42).

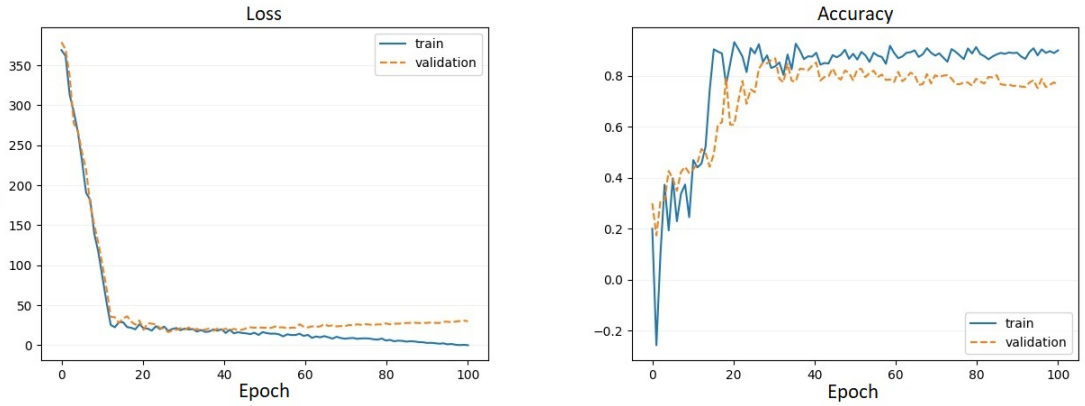

Figure 0.8. The SRMAR-E model training performance plot for the Market-1501 dataset (input image size is 21x42).

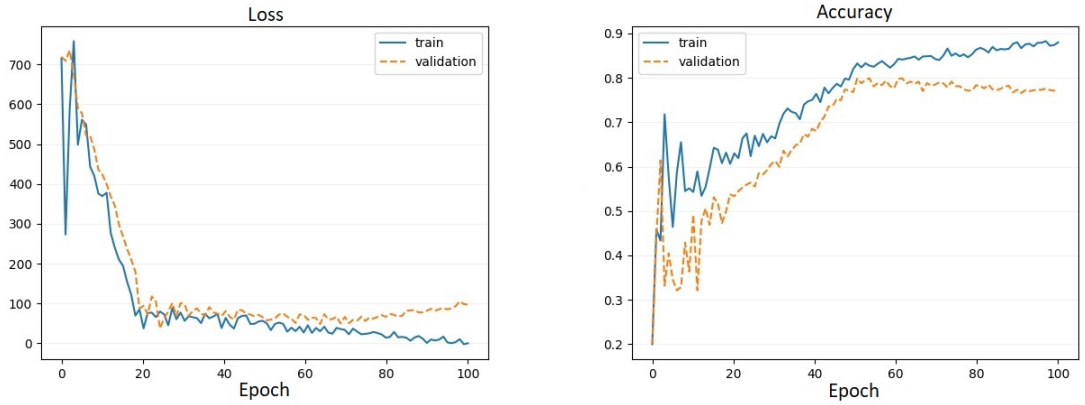

Figure 0.9. Thef SRMAR-D model training performance plot for the DukeMMTC-reID dataset (input image size is 32x64).

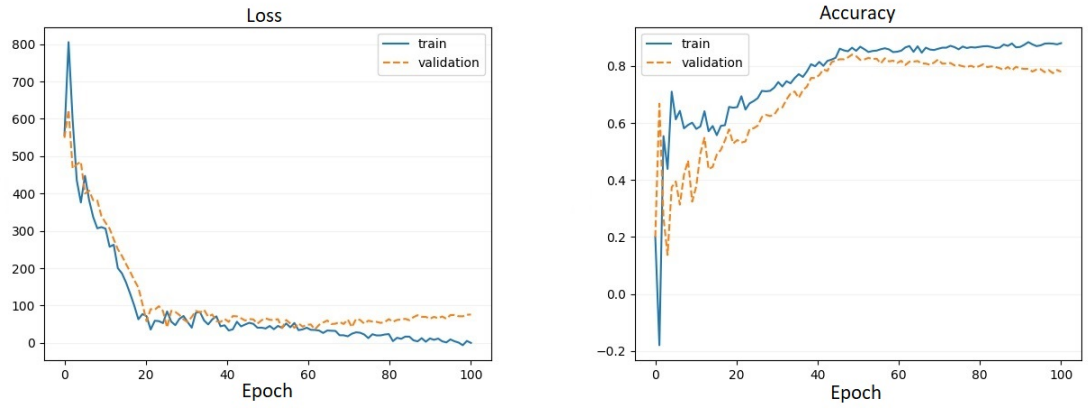

Figure 0.10. The SRMAR-E model training performance plot for the DukeMMTC-reID dataset (input image size is 32x64).

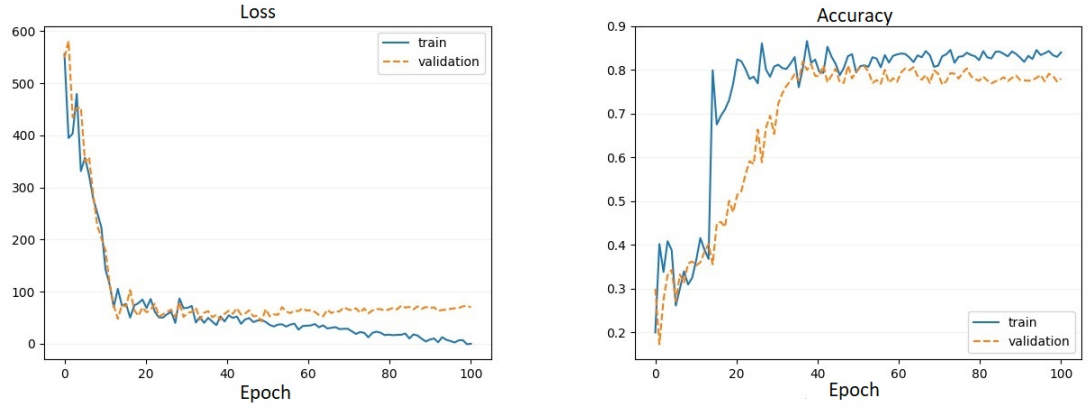

Figure 0.11. The SRMAR-D model training performance plot for the Market-1501 dataset (input image size is 32x64).

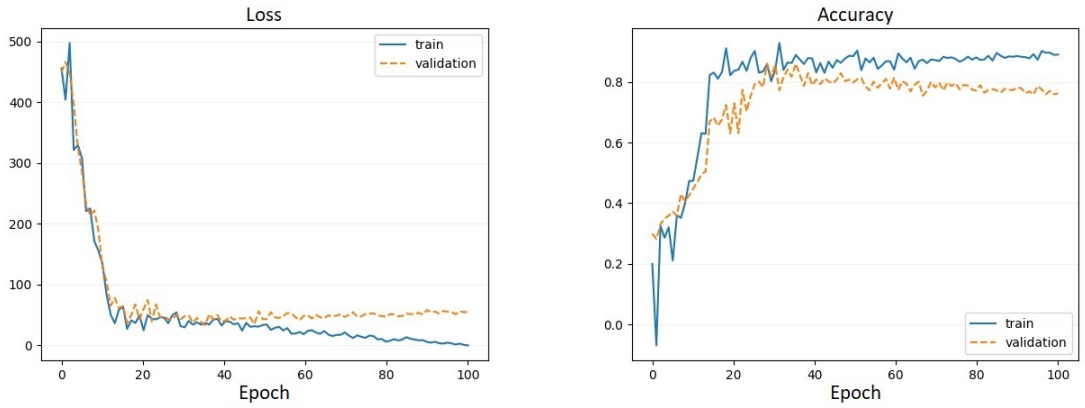

Figure 0.12. The SRMAR-E model training performance plot for the Market-1501 dataset (input image size is 32x64).

Appendix B

Plots of weights from linear combination model

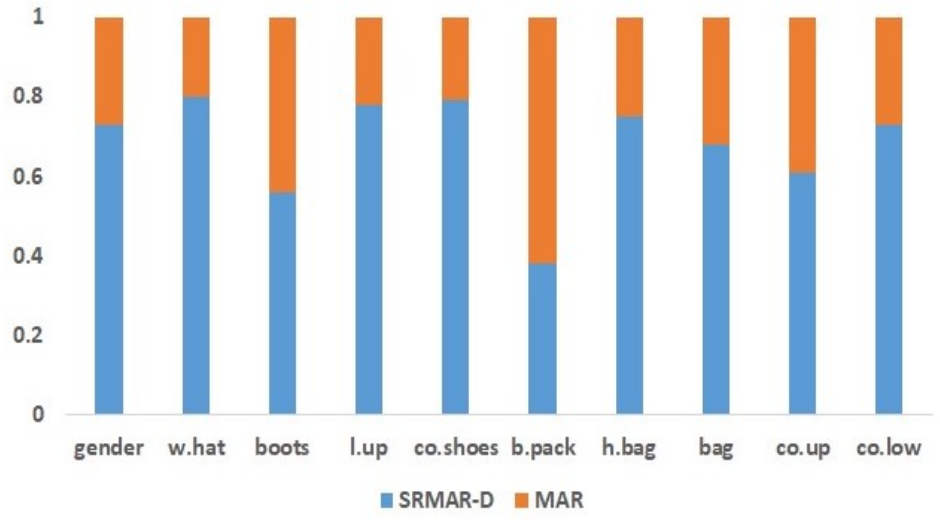

Figure 0.1. The plot of Linear combination model weights concerning SRMAR-D and MAR model getting weights in DukeMTMC-reID. (input sizes 8×16)

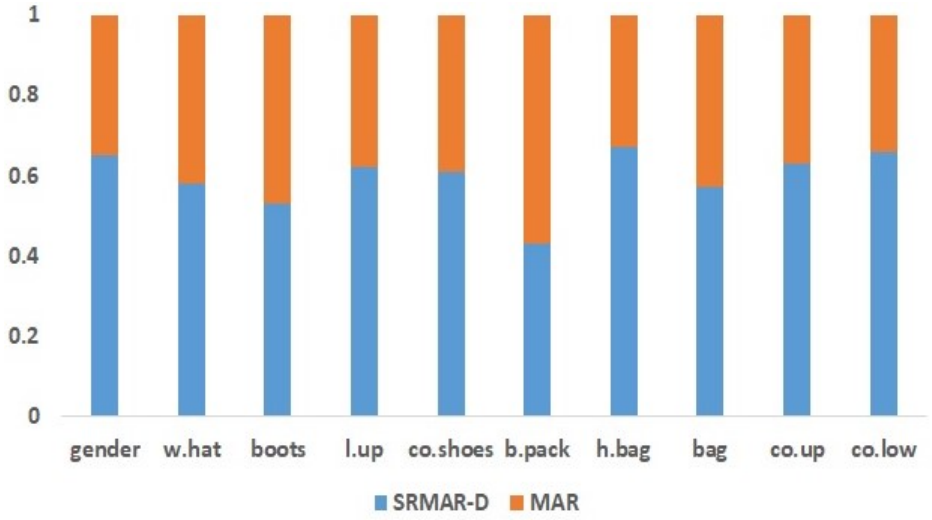

Figure 0.2. The plot of Linear combination model weights concerning SRMAR-D and MAR model getting weights in DukeMTMC-reID. (input sizes 16×32)

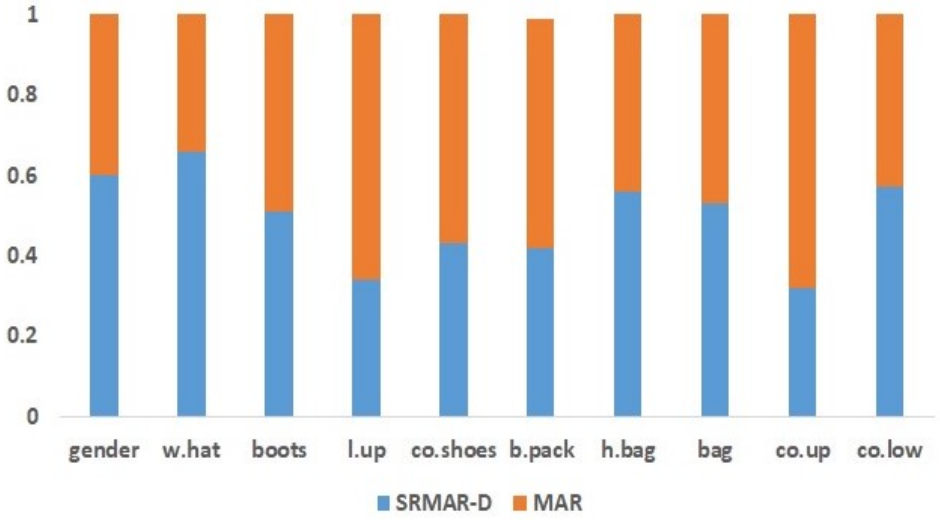

Figure 0.3. The plot of Linear combination model weights concerning SRMAR-D and MAR model getting weights in DukeMTMC-reID. (input sizes 32×64)

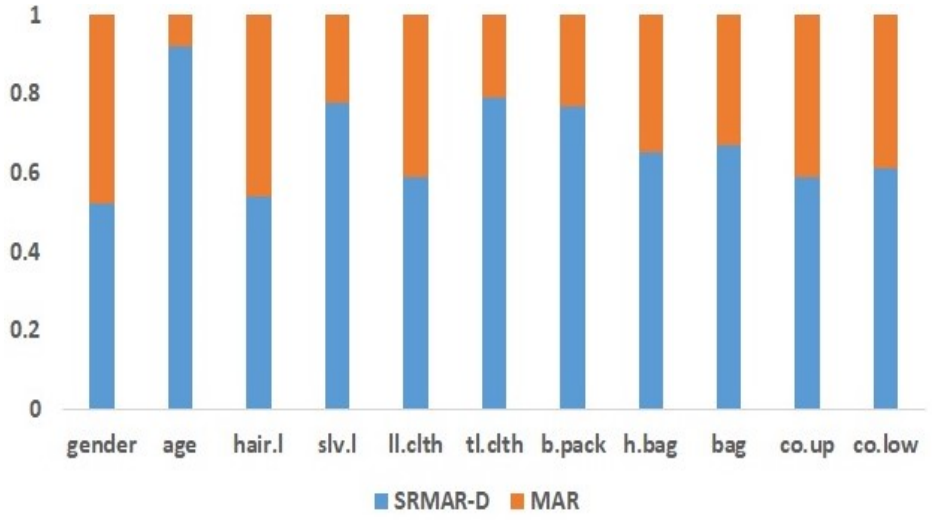

Figure 0.4. The plot of Linear combination model weights concerning SRMAR-D and MAR model getting weights in Market-1501. (input sizes 8×16)

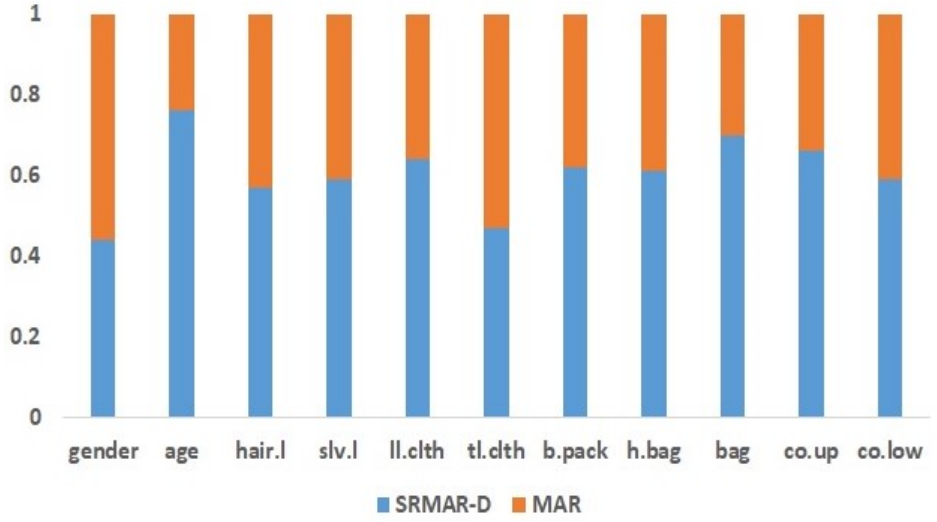

Figure 0.5. The plot of Linear combination model weights concerning SRMAR-D and MAR model getting weights in Market-1501. (input sizes 16×32)

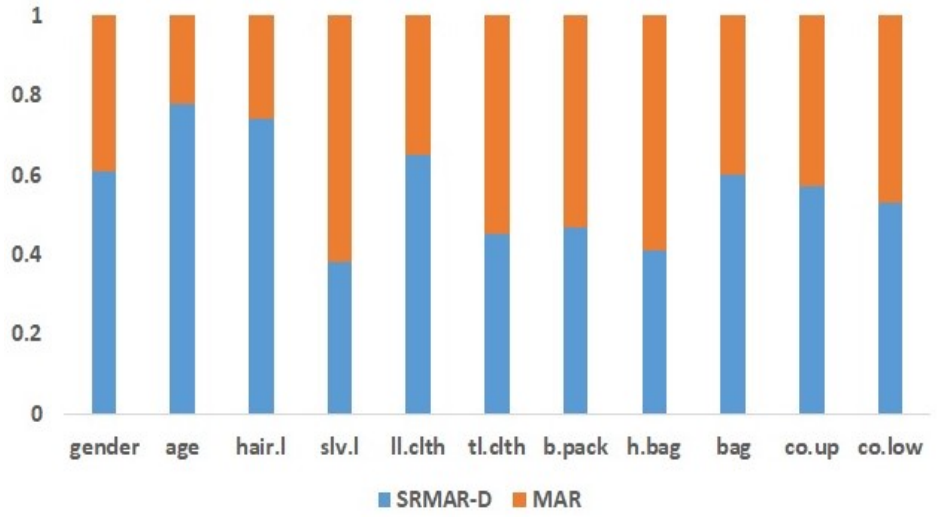

Figure 0.6. The plot of Linear combination model weights concerning SRMAR-D and MAR model getting weights in Market-1501. (input sizes 32×64)

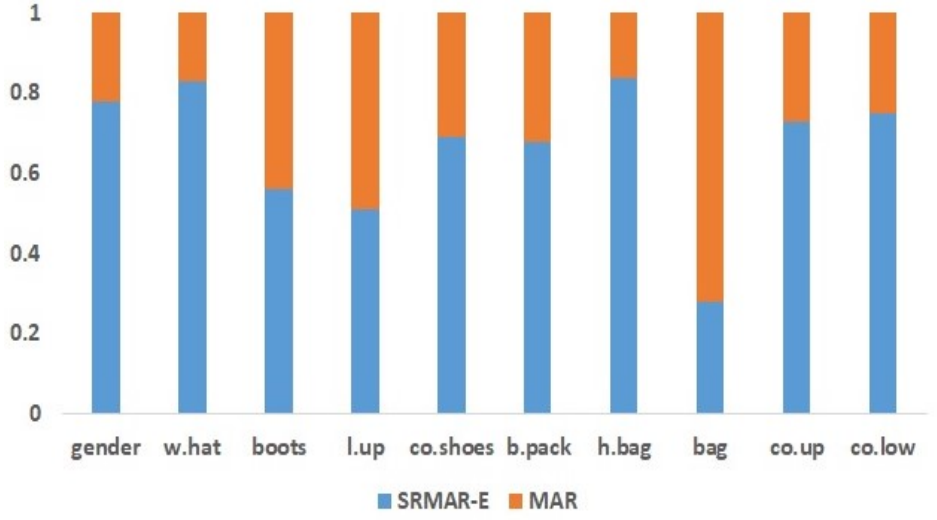

Figure 0.7. The plot of Linear combination model weights concerning SRMAR-E and MAR model getting weights in DukeMTMC-reID. (input sizes 16×32)

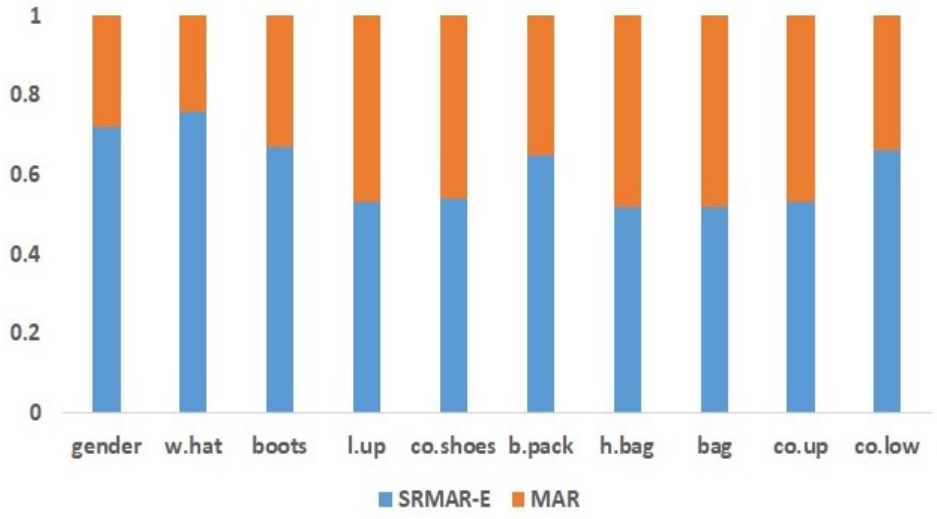

Figure 0.8. The plot of Linear combination model weights concerning SRMAR-E and MAR model getting weights in DukeMTMC-reID. (input sizes 21×42)

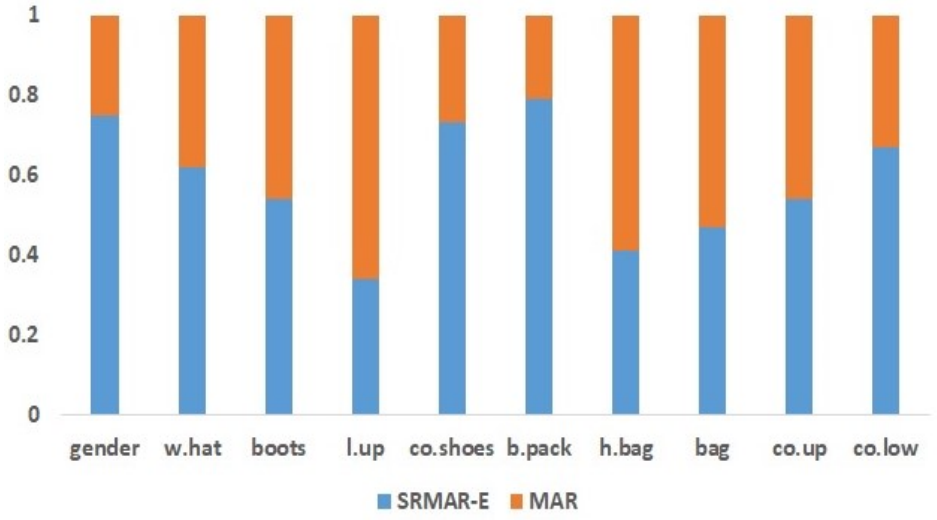

Figure 0.9. The plot of Linear combination model weights concerning SRMAR-E and MAR model getting weights in DukeMTMC-reID. (input sizes 32×64)

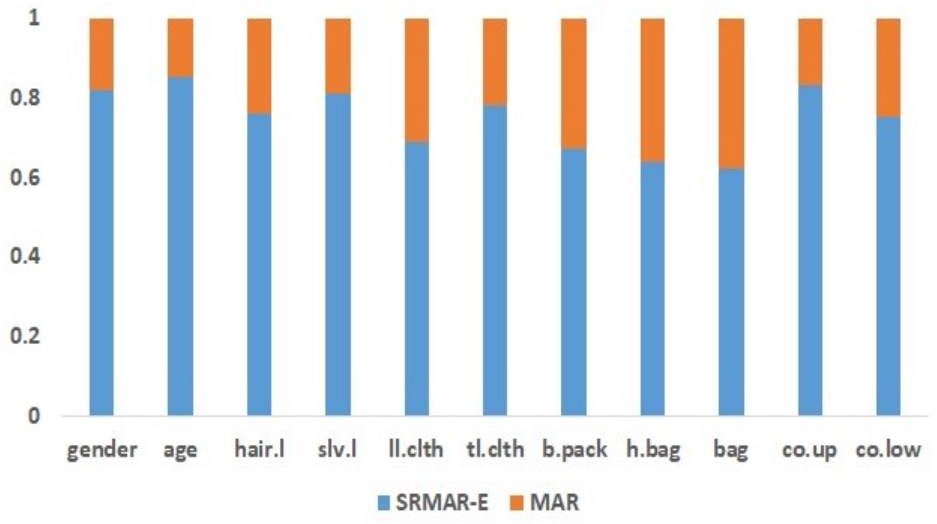

Figure 0.10. The plot of Linear combination model weights concerning SRMAR-E and MAR model getting weights in Market-1501. (input sizes 16×32)

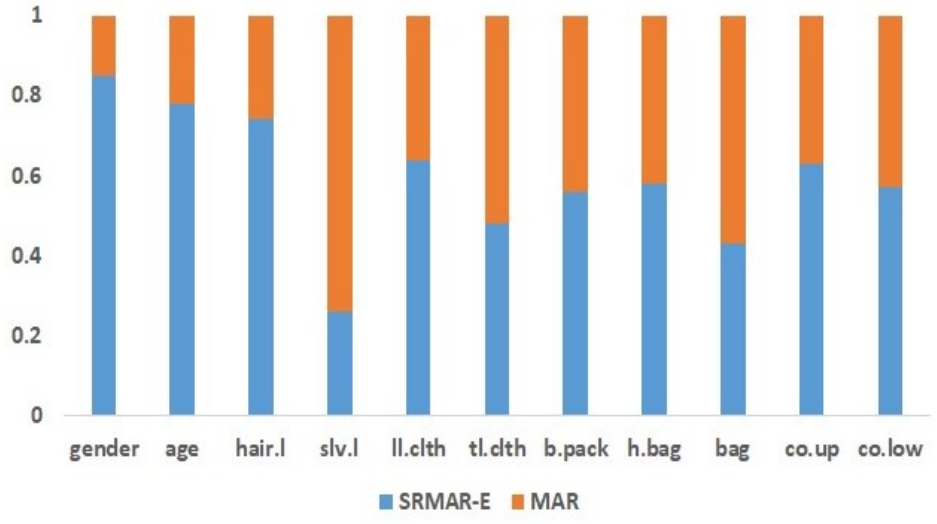

Figure 0.11. The plot of Linear combination model weights concerning SRMAR-E and MAR model getting weights in Market-1501. (input sizes 21×42)

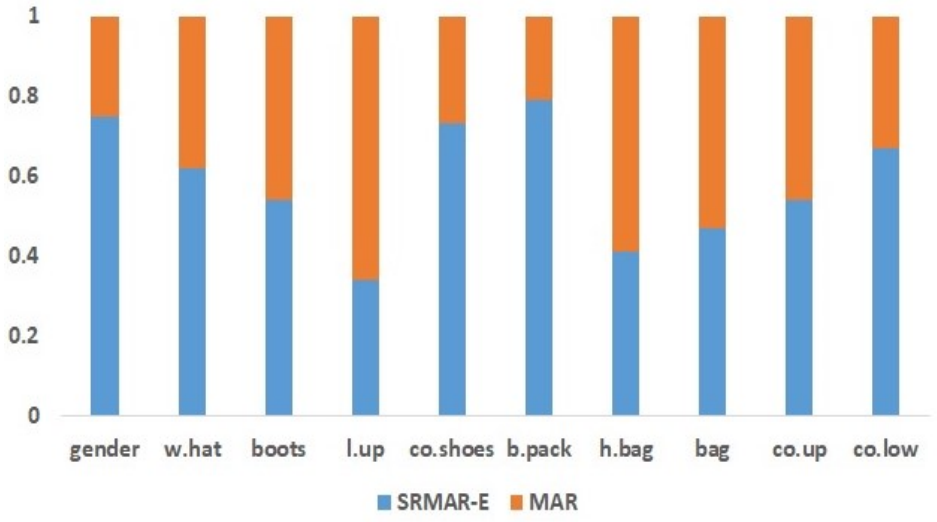

Figure 0.12. The plot of Linear combination model weights concerning SRMAR-E and MAR model getting weights in Market-1501. (input sizes 32×64)

REFERENCES

[1] Liang Zheng, Liyue Shen, Lu Tian, Shengjin Wang, Jingdong Wang, and Qi Tian. Scalable person re-identification: A benchmark. In *Proceedings of the IEEE international conference on computer vision*, pages 1116–1124. **2015**.

[2] Zhedong Zheng, Liang Zheng, and Yi Yang. Unlabeled samples generated by gan improve the person re-identification baseline in vitro. In *Proceedings of the IEEE international conference on computer vision*, pages 3754–3762. **2017**.

[3] Diego Sousa. Google e facebook se manisfestam contra app de reconhecimento facial; entenda, **2020**.

[4] Neeraj Kumar, Alexander Berg, Peter N Belhumeur, and Shree Nayar. Describable visual attributes for face verification and image search. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(10):1962–1977, **2011**.

[5] Behjat Siddiquie, Rogerio S Feris, and Larry S Davis. Image ranking and retrieval based on multi-attribute queries. In *CVPR 2011*, pages 801–808. IEEE, **2011**.

[6] Yutian Lin, Liang Zheng, Zhedong Zheng, Yu Wu, Zhilan Hu, Chenggang Yan, and Yi Yang. Improving person re-identification by attribute and identity learning. *Pattern Recognition*, 95:151–161, **2019**.

[7] IAN NORMAN. A practical guide to creating superresolution photos with photoshop, **2015**.

[8] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, **2012**.

[9] Jin Yamanaka, Shigesumi Kuwashima, and Takio Kurita. Fast and accurate image super resolution by deep cnn with skip connection and network in network.

In *International Conference on Neural Information Processing*, pages 217–225. Springer, **2017**.

[10] Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. Enhanced deep residual networks for single image super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 136–144. **2017**.

[11] Muhammad Haris, Gregory Shakhnarovich, and Norimichi Ukita. Deep back-projection networks for super-resolution. In *CVPR*, pages 1664–1673. **2018**.

[12] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Image super-resolution using deep convolutional networks. *IEEE transactions on pattern analysis and machine intelligence*, 38(2):295–307, **2015**.

[13] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. Accurate image super-resolution using very deep convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1646–1654. **2016**.

[14] Ying Tai, Jian Yang, and Xiaoming Liu. Image super-resolution via deep recursive residual network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3147–3155. **2017**.

[15] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. Deeply-recursive convolutional network for image super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1637–1645. **2016**.

[16] Chao Dong, Chen Change Loy, and Xiaoou Tang. Accelerating the super-resolution convolutional neural network. In *European conference on computer vision*, pages 391–407. Springer, **2016**.

[17] Wenzhe Shi, Jose Caballero, Ferenc Huszár, Johannes Totz, Andrew P Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network.

In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1874–1883. **2016**.

[18] Radu Timofte, Eirikur Agustsson, Luc Van Gool, Ming-Hsuan Yang, and Lei Zhang. Ntire 2017 challenge on single image super-resolution: Methods and results. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 114–125. **2017**.

[19] Wei-Sheng Lai, Jia-Bin Huang, Narendra Ahuja, and Ming-Hsuan Yang. Deep laplacian pyramid networks for fast and accurate super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 624–632. **2017**.

[20] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. In *CVPR*, pages 4681–4690. **2017**.

[21] Marti A. Hearst, Susan T Dumais, Edgar Osuna, John Platt, and Bernhard Scholkopf. Support vector machines. *IEEE Intelligent Systems and their applications*, 13(4):18–28, **1998**.

[22] Yubin Deng, Ping Luo, Chen Change Loy, and Xiaoou Tang. Pedestrian attribute recognition at far distance. In *Proceedings of the 22nd ACM international conference on Multimedia*, pages 789–792. **2014**.

[23] Dangwei Li, Xiaotang Chen, and Kaiqi Huang. Multi-attribute learning for pedestrian attribute recognition in surveillance scenarios. In *2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR)*, pages 111–115. IEEE, **2015**.

[24] Jianqing Zhu, Shengcai Liao, Dong Yi, Zhen Lei, and Stan Z Li. Multi-label cnn based pedestrian attribute learning for soft biometrics. In *2015 International Conference on Biometrics (ICB)*, pages 535–540. IEEE, **2015**.

[25] Ding-Xuan Zhou. Theory of deep convolutional neural networks: Downsampling. *Neural Networks*, 124:319–327, **2020**.

[26] Sin-Ye Jhong, Po-Yen Tseng, Natnuntnita Siriphockpirom, Chih-Hsien Hsia, Ming-Shih Huang, Kai-Lung Hua, and Yung-Yao Chen. An automated biometric identification system using cnn-based palm vein recognition. In *2020 International Conference on Advanced Robotics and Intelligent Systems (ARIS)*, pages 1–6. IEEE, **2020**.

[27] Adil Al-Azzawi, Anes Ouadou, Highsmith Max, Ye Duan, John J Tanner, and Jianlin Cheng. Deepcryopicker: fully automated deep neural network for single protein particle picking in cryo-em. *BMC bioinformatics*, 21(1):1–38, **2020**.

[28] Tao Wang, Changhua Lu, Mei Yang, Feng Hong, and Chun Liu. A hybrid method for heartbeat classification via convolutional neural networks, multilayer perceptrons and focal loss. *PeerJ Computer Science*, 6:e324, **2020**.

[29] Guoqing Li, Meng Zhang, Jiaojie Li, Feng Lv, and Guodong Tong. Efficient densely connected convolutional neural networks. *Pattern Recognition*, 109:107610, **2021**.

[30] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, **1998**.

[31] Yann Lecun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. In *Proceedings of the IEEE*, pages 2278–2324. **1998**.

[32] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *IEEE Conference on Computer Vision and Pattern Recognition*, **2014**.

[33] C. Szegedy, , , P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–9. **2015**. ISSN 1063-6919. doi:10.1109/CVPR.2015.7298594.

[34] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778. **2016**. doi:10.1109/CVPR.2016.90.

[35] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. pages 2261–2269, **2020**.

[36] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, **2009**.

[37] Vittorio Ferrari and Andrew Zisserman. Learning visual attributes. *Advances in neural information processing systems*, 20:433–440, **2007**.

[38] Daniel A Vaquero, Rogerio S Feris, Duan Tran, Lisa Brown, Arun Hampapur, and Matthew Turk. Attribute-based people search in surveillance environments. In *2009 workshop on applications of computer vision (WACV)*, pages 1–8. IEEE, **2009**.

[39] Gary B Huang, Marwan Mattar, Tamara Berg, and Eric Learned-Miller. Labeled faces in the wild: A database forstudying face recognition in unconstrained environments. In *Workshop on faces in'Real-Life'Images: detection, alignment, and recognition*. **2008**.

[40] James Petterson, Tibério S Caetano, et al. Reverse multi-label learning. In *NIPS*, volume 1, page 7. **2010**.

[41] Matthieu Guillaumin, Thomas Mensink, Jakob Verbeek, and Cordelia Schmid. Tagprop: Discriminative metric learning in nearest neighbor models for image

auto-annotation. In *2009 IEEE 12th international conference on computer vision*, pages 309–316. IEEE, **2009**.

[42] Thorsten Joachims. Optimizing search engines using clickthrough data. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 133–142. **2002**.

[43] Yoav Freund, Raj Iyer, Robert E Schapire, and Yoram Singer. An efficient boosting algorithm for combining preferences. *Journal of machine learning research*, 4(Nov):933–969, **2003**.

[44] Quoc Le and Alexander Smola. Direct optimization of ranking measures. *arXiv preprint arXiv:0704.3359*, **2007**.

[45] Ryan Layne, Timothy M Hospedales, Shaogang Gong, and Q Mary. Person re-identification by attributes. In *Bmvc*, volume 2, page 8. **2012**.

[46] Douglas Gray, Shane Brennan, and Hai Tao. Evaluating appearance models for recognition, reacquisition, and tracking. In *Proc. IEEE international workshop on performance evaluation for tracking and surveillance (PETS)*, volume 3, pages 1–7. Citeseer, **2007**.

[47] Wei-Shi Zheng, Shaogang Gong, and Tao Xiang. Associating groups of people. In *BMVC*, volume 2, pages 1–11. **2009**.

[48] Subhransu Maji, Alexander C Berg, and Jitendra Malik. Classification using intersection kernel support vector machines is efficient. In *2008 IEEE conference on computer vision and pattern recognition*, pages 1–8. IEEE, **2008**.

[49] Chi Su, Fan Yang, Shiliang Zhang, Qi Tian, Larry S Davis, and Wen Gao. Multi-task learning with low rank attribute embedding for person re-identification. In *Proceedings of the IEEE international conference on computer vision*, pages 3739–3747. **2015**.

[50] Rich Caruana. Algorithms and applications for multitask learning. In *ICML*, pages 87–95. Citeseer, **1996**.

[51] Taiqing Wang, Shaogang Gong, Xiatian Zhu, and Shengjin Wang. Person re-identification by video ranking. In *European conference on computer vision*, pages 688–703. Springer, **2014**.

[52] Martin Hirzer, Csaba Beleznai, Peter M Roth, and Horst Bischof. Person re-identification by descriptive and discriminative classification. In *Scandinavian conference on Image analysis*, pages 91–102. Springer, **2011**.

[53] Alina Bialkowski, Simon Denman, Sridha Sridharan, Clinton Fookes, and Patrick Lucey. A database for person re-identification in multi-camera surveillance networks. In *2012 International Conference on Digital Image Computing Techniques and Applications (DICTA)*, pages 1–8. IEEE, **2012**.

[54] Lubomir Bourdev, Subhransu Maji, and Jitendra Malik. Describing people: A poselet-based approach to attribute classification. In *2011 International Conference on Computer Vision*, pages 1543–1550. IEEE, **2011**.

[55] Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. A deep convolutional activation feature for generic visual recognition. *UC Berkeley & ICSI, Berkeley, CA, USA*, 1.

[56] Jianqing Zhu, Shengcai Liao, Zhen Lei, Dong Yi, and Stan Li. Pedestrian attribute classification in surveillance: Database and evaluation. In *Proceedings of the IEEE international conference on computer vision workshops*, pages 331–338. **2013**.

[57] Chunxiao Liu, Shaogang Gong, Chen Change Loy, and Xinggang Lin. Person re-identification: What features are important? In *European Conference on Computer Vision*, pages 391–401. Springer, **2012**.

[58] Patrick Sudowe, Hannah Spitzer, and Bastian Leibe. Person attribute recognition with a jointly-trained holistic cnn model. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 87–95. **2015**.

[59] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the 22nd ACM international conference on Multimedia*, pages 675–678. **2014**.

[60] Gaurav Sharma and Frederic Jurie. Learning discriminative spatial representation for image classification. In *BMVC 2011-British Machine Vision Conference*, pages 1–11. BMVA Press, **2011**.

[61] Yining Li, Chen Huang, Chen Change Loy, and Xiaoou Tang. Human attribute recognition by deep hierarchical contexts. In *European Conference on Computer Vision*, pages 684–700. Springer, **2016**.

[62] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448. **2015**.

[63] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778. **2016**.

[64] Yuxuan Shi, Hefei Ling, Lei Wu, Jialie Shen, and Ping Li. Learning refined attribute-aligned network with attribute selection for person re-identification. *Neurocomputing*, 402:124–133, **2020**.

[65] Jingjing Wu, Hao Liu, Jianguo Jiang, Meibin Qi, Bo Ren, Xiaohong Li, and Yashen Wang. Person attribute recognition by sequence contextual relation learning. *IEEE Transactions on Circuits and Systems for Video Technology*, 30(10):3398–3412, **2020**.

[66] Jianchao Yang and Thomas Huang. Image super-resolution: Historical overview and future challenges. In *Super-resolution imaging*, pages 1–34. CRC Press, **2017**.

[67] R Tsai. Multiframe image restoration and registration. *Advance Computer Visual and Image Processing*, 1:317–339, **1984**.

[68] Michal Irani and Shmuel Peleg. Super resolution from image sequences. In *[1990] Proceedings. 10th International Conference on Pattern Recognition*, volume 2, pages 115–120. IEEE, **1990**.

[69] Jianchao Yang, John Wright, Thomas S Huang, and Yi Ma. Image super-resolution via sparse representation. *IEEE transactions on image processing*, 19(11):2861–2873, **2010**.

[70] Zhaowen Wang, Ding Liu, Jianchao Yang, Wei Han, and Thomas Huang. Deep networks for image super-resolution with sparse prior. In *Proceedings of the IEEE international conference on computer vision*, pages 370–378. **2015**.

[71] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Learning a deep convolutional network for image super-resolution. In *European conference on computer vision*, pages 184–199. Springer, **2014**.

[72] Dengwen Zhou, Ran Duan, Lijuan Zhao, and Xiaoliang Chai. Single image super-resolution reconstruction based on multi-scale feature mapping adversarial network. *Signal Processing*, 166:107251, **2020**.

[73] Xintao Wang, Ke Yu, Shixiang Wu, Jinjin Gu, Yihao Liu, Chao Dong, Yu Qiao, and Chen Change Loy. Esrgan: Enhanced super-resolution generative adversarial networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 0–0. **2018**.

[74] Pourya Shamsolmoali, Masoumeh Zareapoor, Ruili Wang, Deepak Kumar Jain, and Jie Yang. G-ganisr: Gradual generative adversarial network for image super resolution. *Neurocomputing*, 366:140–153, **2019**.

[75] Chuantao Fang, Yu Zhu, Lei Liao, and Xiaofeng Ling. Tsrgan: Real-world text image super-resolution based on adversarial learning and triplet attention. *Neurocomputing*, 455:88–96, **2021**.

[76] Jijun He, Jinjin Zheng, Yuan Shen, Yutang Guo, and Hongjun Zhou. Facial image synthesis and super-resolution with stacked generative adversarial network. *Neurocomputing*, 402:359–365, **2020**.

[77] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. PMLR, **2015**.

[78] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034. **2015**.

[79] Marco Bevilacqua, Aline Roumy, Christine Guillemot, and Marie Line Alberi-Morel. Low-complexity single-image super-resolution based on nonnegative neighbor embedding. **2012**.

[80] Roman Zeyde, Michael Elad, and Matan Protter. On single image scale-up using sparse-representations. In *International conference on curves and surfaces*, pages 711–730. Springer, **2010**.

[81] David Martin, Charless Fowlkes, Doron Tal, and Jitendra Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*, volume 2, pages 416–423. IEEE, **2001**.

[82] Min Lin, Qiang Chen, and Shuicheng Yan. Network in network. *arXiv preprint arXiv:1312.4400*, **2013**.

[83] Pablo Arbelaez, Michael Maire, Charless Fowlkes, and Jitendra Malik. Contour detection and hierarchical image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 33(5):898–916, **2010**.

[84] Seungjun Nah, Tae Hyun Kim, and Kyoung Mu Lee. Deep multi-scale convolutional neural network for dynamic scene deblurring. In *CVPR*, pages 3883–3891. **2017**.

[85] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *Thirty-first AAAI conference on artificial intelligence*. **2017**.

[86] Pierre Charbonnier, Laure Blanc-Feraud, Gilles Aubert, and Michel Barlaud. Two deterministic half-quadratic regularization algorithms for computed imaging. In *Proceedings of 1st International Conference on Image Processing*, volume 2, pages 168–172. IEEE, **1994**.

[87] C. A. Knox Lovell C. L. Kenneth and S. Thore. Chance-constrained data envelopment analysis. *Managerial and Decision Economics*, 14(6):541–554, **1993**.

[88] Gerald Gamrath et al. The SCIP Optimization Suite 7.0. ZIB-Report 20-10, Zuse Institute Berlin, **2020**.

[89] Shiv Ram Dubey, Soumendu Chakraborty, Swalpa Kumar Roy, Snehasis Mukherjee, Satish Kumar Singh, and Bidyut Baran Chaudhuri. Diffgrad: an optimization method for convolutional neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, **2019**.