Bachelor of Science in Computer Science & Engineering

Pedestrian Attribute Recognition Based on Custom Designed CNN Model

by

Saadman Sakib

ID: 1504026

Department of Computer Science & Engineering

Chittagong University of Engineering & Technology (CUET)

Chattogram-4349, Bangladesh.

May, 2021

Pedestrian Attribute Recognition Based on Custom Designed CNN Model

Submitted in partial fulfilment of the requirements for
Degree of Bachelor of Science
in Computer Science & Engineering

by

Saadman Sakib

ID: 1504026

Supervised by

Dr. Kaushik Deb

Professor

Department of Computer Science & Engineering

Chittagong University of Engineering & Technology (CUET)

Chattogram-4349, Bangladesh.

The thesis titled '**Pedestrian Attribute Recognition Based on Custom Designed CNN Model'** submitted by ID: 1504026, Session 2019-2020 has been accepted as satisfactory in fulfilment of the requirement for the degree of Bachelor of Science in Computer Science & Engineering to be awarded by the Chittagong University of Engineering & Technology (CUET).

Board of Examiners

Chairman(Supervisor)

Dr. Kaushik Deb
Professor
Department of Computer Science & Engineering
Chittagong University of Engineering & Technology (CUET)

Member (Ex-Officio)

Dr. Md. Mokammel Haque
Professor & Head
Department of Computer Science & Engineering
Chittagong University of Engineering & Technology (CUET)

Member (External)

Dr. Muhammad Ibrahim Khan
Professor
Department of Computer Science & Engineering
Chittagong University of Engineering & Technology (CUET)

Declaration of Originality

This is to certify that I am the sole author of this thesis and that neither any part of this thesis nor the whole of the thesis has been submitted for a degree to any other institution.

I certify that, to the best of my knowledge, my thesis does not infringe upon anyone's copyright nor violate any proprietary rights and that any ideas, techniques, quotations, or any other material from the work of other people included in my thesis, published or otherwise, are fully acknowledged in accordance with the standard referencing practices. I am also aware that if any infringement of anyone's copyright is found, whether intentional or otherwise, I may be subject to legal and disciplinary action determined by Dept. of CSE, CUET.

I hereby assign every rights in the copyright of this thesis work to Dept. of CSE, CUET, who shall be the owner of the copyright of this work and any reproduction or use in any form or by any means whatsoever is prohibited without the consent of Dept. of CSE, CUET.

Signature of the candidate

Date: 21/05/2021

Acknowledgements

First and foremost, I am grateful to Allah, who has helped me to successfully complete this thesis. Following that, I would like to express my heartfelt gratitude to my honourable project supervisor, Dr. Kaushik Deb, Professor of Computer Science & Engineering Department, Chittagong University of Engineering & Technology, for his insightful suggestions, encouraging words, and sincere guidance during my thesis work. I'd like to express my heartfelt appreciation to all of the department's esteemed faculty members. I'd like to express my gratitude to all of my friends as well as the department's staffs for their helpful advice and assistance. Finally, I'd like to express my gratitude to my parents for their unwavering love and support during my academic career.

Abstract

Pedestrian attribute recognition has recently created a significant impact because of its soft bio-metric property to recognise individuals. Since there is less data and some difficult factors, such as viewpoints and lighting conditions, existing approaches recognise pedestrian attributes with less accuracy. Also, in a real-world scenario, an image might contain multiple pedestrians. So recognising the attributes of the pedestrians is a challenging task. Our main goal is to customise a CNN architecture to predict the pedestrian attributes better. In our work, we proposed a framework that can work in a real-world scenario for detecting multiple pedestrians with their attributes. Secondly, a brief analysis of the RAP dataset used in our work is shown. Thirdly, a comparative analysis of top-performing pre-trained models on RAP dataset with Transfer Learning approach is performed for selecting primary CNN architecture. Finally, extensive experiment on the primary architecture is carried out in order to achieve better results. Our proposed method outperformed existing methods by means of accuracy.

Keywords: Transfer learning, CNN, Mask-RCNN, Oversampling, Spatial features, Confusion matrix, Feature map, Evaluation metric

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Introduction

The modern world produces vast quantities of data regularly as a result of technological advances. In the field of computer vision, image data makes a significant contribution. Image data and advanced techniques are necessary for performing image recognition, image classification, or object detection. Image recognition aims to automatically recognise an object or a group of objects without the need for human interference. This process is focused on the semantic content of an image. The modern world is reaching a point where human interference is no longer necessary. Each and every job will be completed automatically. This major shift involves image recognition.

Pedestrian attribute recognition, in this context, refers to the classification of pedestrian attributes in surveillance scenarios based on semantic features. This task aims to explore the pedestrian attribute dataset and use various techniques to identify pedestrian attributes. A comparison of these methods will also be provided. This is a unique approach to the subject.

Machine learning or deep learning techniques are now needed for automation tasks since they achieve better results than other approaches. Deep learning techniques are used instead of the previously popular handcraft methods for image processing. The main explanation for this is that a deep learning-based model can extract features automatically, while handcrafted approaches need manual intervention. For image processing tasks, DNN or Deep Neural Network-based models can be extremely complex. These DNN models also have a wide number of parameters. Image recognition can now be done in real-time thanks to advances in

computing power, but achieving better results is still a challenge. So, pedestrian attribute recognition with better results is a very challenging task.

The proposed Pedestrian Attribute Recognition Based on Custom Designed CNN Model and the challenges encountered in the process will be addressed in this chapter. This chapter will also discuss the thesis topic's inspiration and implementation.

1.2 Pedestrian Attribute Recognition Framework

Pedestrian images have spatial correlations that are used to identify pedestrian attributes. The most challenging aspect of object recognition is capturing spatial correlations. Image features is another term for this. We would be able to achieve good results if we can effectively capture image features. As a result, pedestrian attribute recognition is a form of object recognition that has been expanded.

An essential method for extracting features from images is the Convolutional Neural Network. Because of the usage of multiple function extraction stages that can acquire representations from data automatically, Deep CNN has a high learning ability. The accessibility of huge amounts of data as well as technological advancements have accelerated CNN science, and recently, several intriguing deep CNN architectures have been published as proposed in [1].

However, architectural advancements are responsible for the deep CNN's dramatic increase in representational ability. Exploiting spatial and channel detail, depth and width of the design, and multi-path information processing have all received a lot of publicity recently. Similarly, the use of a layer block as a structural unit is gaining popularity.

Some pre-processing tasks are needed to scale the data at the start of the pedestrian attribute recognition approach. To summarise, 1.1 depicts the block diagram of the pedestrian attribute recognition framework.

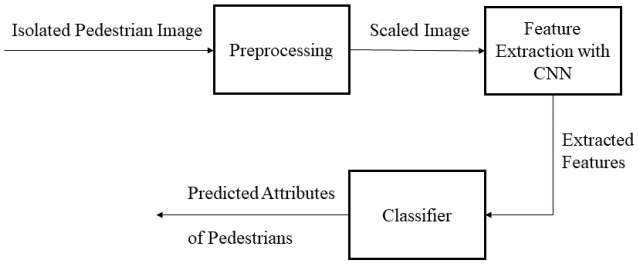


Figure 1.1: Block Diagram of Pedestrian Attribute Recognition Framework.

1.3 Challenges

Deep Learning models face particular challenges when it comes to recognising pedestrian characteristics. The process of extracting spatial information from images and using it to classify attributes is known as the pedestrian attribute recognition task. Following is a list of the main challenges:

- Recognising characteristics by capturing dynamic spatial features
- Developing an optimal model that can effectively recognise pedestrian attributes
- Computational complexity as a result of working with a large dataset with a lot of model parameters
- Different camera orientation
- Due to different clothing appearances, large intra-class differences.
- Uneven illumination conditions
- Occlusion in an image
- Low-resolution image from surveillance cameras
- Varied distance

1.4 Applications

Gender, age, shirt, pant, and other pedestrian attributes have been used as soft biometric traits in the surveillance sector and have recently received a lot of attention. These attributes can help to catch criminals in real-time surveillance scenario. As a result, it can be used to fight crime. It also has a lot of promise in terms of smart video monitoring [2] and video-based business intelligence. PAR also has other application areas which can benefit people if properly explored. The following areas are:

- Person re-identification
- Person recognition
- Public security
- Suspicious person identification

1.5 Motivation

Deep learning has provided a massive boost to the field of computer vision, which is already advancing at a breakneck rate. Many new applications of computer vision methods have been implemented as a result of deep learning, and they are now part of our daily lives. The essential part of the modern world is to reduce human interference. Every day, new deep learning techniques are implemented, with many of them producing substantial results. We may infer from this that deep learning has a large research field as well as the potential to solve complex problems. Pedestrian attribute recognition task with deep learning methods is quite a novel approach. Though there are some existing works, there is still room for improvement. The key reasons motivating this work are given below:

- Great research opportunity
- Opportunity to reduce the crime rate
- Large application scope for real-world scenarios as shown in 1.4
- Overcome demerits of existing works

1.6 Contribution of the thesis

The aim of a thesis or research paper is to contribute to the advancement of modern science by supplementing previous achievements with proper logic and facts. The main goal of this project is to create a pedestrian attribute recognition framework that can recognise attributes. The following are some of the main goals and results of this project:

- A framework that can work in a real-world scenario for detecting multiple pedestrians with their attributes
- Comparing among various CNN architectures with additional layers using Transfer Learning approach for selecting the primary architecture
- Proposing a CNN architecture by experimenting with the primary architecture
- Analysing the dataset with more attributes and data balancing techniques

1.7 Thesis Organisation

The rest of this thesis report is organised as follows:

- Chapter 2 gives a summary of previous research works in the field of Pedestrian Attribute Recognition
- Chapter 3 describes the proposed methodology for the Pedestrian Attribute Recognition task in detail
- Chapter 4 explains the details of the working dataset and represents the analysis of the performance of the proposed framework
- Chapter 5 contains the overall summary of this thesis work and provides some future recommendations as well

1.8 Conclusion

An overview of the Pedestrian Attribute Recognition system is given in this chapter. This chapter also includes a rundown of the Pedestrian Attribute Recognition Framework, in addition to the challenges. This is also where the inspiration for this work is stated. The history and current state of the problem will be addressed in the following chapter.

Chapter 2

Literature Review

2.1 Introduction

The popularity of Deep Learning was addressed in chapter 1. Using deep learning to recognise pedestrian attributes is also a novel approach. For the researchers, this has provided a large research field. The development of Deep Learning techniques, including a number of CNN architectures, has sparked interest in the task of recognising pedestrian attributes. The challenge has been made simpler by the variety and availability of datasets.

Following a brief overview of the previous study, this chapter addresses various feature representation and classification methods used by various researchers using Deep Learning approaches, as well as the results of these studies on various datasets.

2.2 Related Literature Review on Pedestrian Attribute Recognition

2.2.1 Feature Extraction using Traditional Methods

The multi-label classification issue of pedestrian attribute recognition has been widely used in the retrieval and re-identification of individuals. Attribute analysis is becoming more common as it can be used to infer high-level semantic details. Traditional approaches to solving this problem have been used in previous studies.

Rather than creating a particular feature by hand to solve the problem, Gray et al. in [3] demonstrated how the AdaBoost algorithm could be used to learn a

discriminative recognition model as well as an object class specific representation. Prosser et al. create an Ensemble RankSVM to solve the scalability problem that plagues current SVM-based ranking methods in [4]. This new model uses considerably less memory, making it much more flexible while still retaining high performance.

Layne et al. proposed a novel re-identification approach that learns a range and weighting of mid-level semantic attributes to identify individuals in [5] & [6]. The model learns an attribute-centric, parts-based feature representation in particular. They also demonstrated how mid-level "semantic attributes" for individual definition could be computed.

Later, Deng et al. [7] published a large-scale dataset that makes learning robust attribute detectors with strong generalisation efficiency much easier. In addition, the benchmark output was presented using an SVM-based method, and an alternative approach was suggested that uses the background of neighbouring pedestrian images to enhance attribute inference.

Li et al. published a Richly Annotated Pedestrian (RAP) dataset with long-term data collection from real multi-camera surveillance scenarios in [8], where data samples are annotated with not only fine-grained human attributes but also environmental and contextual variables. They also looked at how various environmental and contextual variables influenced pedestrian attribute recognition.

2.2.2 Feature Extraction using Deep Learning Methods

For tuning a deep learning architecture in an individual re-identification task, attribute identification is becoming increasingly common. In video monitoring and video forensics, recognising pedestrian characteristics is a major issue. Traditional approaches build handcrafted features for each pedestrian characteristic, assuming they are independent. Early research focused on datasets of pedestrian attributes that were relatively thin. In video surveillance, Daniel et al. [9] suggested a part-based feature representation for human features such as facial hair, eyewear, and clothing colour. However, as opposed to current methods, the results are dismal.

Deep learning methods, such as deep Convolutional Neural Networks (CNN), on the other hand, have shown superior results in a variety of computer vision tasks. DeepSAR (where the attributes are considered independent) and DeepMAR (where the attributes are considered dependent) are two CNN-based attribute recognition methods proposed by D. Li et al. [10]. (where the attributes are considered as a dependent). They outperformed the state-of-the-art MRFr2 system. P. Sudowe et al. [11] suggested a method for jointly training a CNN model for all attributes that can reap the benefits of attribute dependencies, using only the image as input and no other external pose, part, or context information. Their multidisciplinary approach is special. CNN outperforms HATDB and Berkeley Attributes of People on two publicly accessible attribute datasets.

J. Zhu et al. [12] proposed a system in which an image is divided into 15 overlapping image parts, and each component is fed into a multi-label Convolutional Neural Network (MLCNN) at the same time, in contrast to current methods that presume attribute independence during prediction. On the Vipers and GRID datasets, experimental findings show that the model performs better. Even so, it does not demonstrate a high level of efficiency.

The network AlexNet is fine-tuned in [13] by T. Matsukawa et al. to make it encode an image into a discriminative function based on the corresponding attributes. Gender, age, luggage-style, upper body clothing type, upper body colour, lower body colour, and lower body clothing type are among the seven categories of attributes considered. The results of the experiment show that fine-tuning using attribute information enhances re-identification accuracy. However, the results are still unsatisfactory. W. Fang et al. [14] suggested a grouping strategy approach based on a fine-tuned VGG-16 structure, in which the attributes are first grouped and then fed into a pre-trained VGG-16 model with minor modifications for improved performance. A custom-tailored CNN network for a particular task will outperform a pre-trained network like AlexNet, ResNet, according to L. Kurnianggoro et al. in [15].

Based on the affinity between pre-extracted proposals and attribute positions, assign attribute-specific weights to local features P. Liu et al. proposed a novel

Localisation Directed Network in [16], as current approaches have difficulty localising the areas corresponding to different attributes. K. Han et al. in [17] proposed an attribute aware pooling algorithm that explores and exploits the association between attributes for the pedestrian attribute recognition challenge, extending the strength of deep convolutional neural networks (CNNs) to the pedestrian attribute recognition issue. Z. Tan et al. in [18] suggest joint learning of three attention mechanisms, namely Parsing attention, Label attention, and Spatial attention, for pedestrian attribute analysis in order to select relevant and discriminative regions or pixels against variations.

Y. Li et al. proposed an attention-based neural network consisting of a convolutional neural network, channel attention (CAtt), and convolutional long short-term memory (ConvLSTM) (CNN-CAtt) in [19] to solve the problem of current pedestrian attribute recognition methods not performing well because they ignored the relationship between pedestrian attributes and spatial information. H. Zeng et al. proposed a novel Co-Attentive Sharing (CAS)module in [20] to address the same problem, which extracts discriminative channels and partial regions for more efficient feature sharing in multi-task learning.

2.3 Conclusion

This chapter contains a thorough summary of the literature review. For clarity, the topic is split into two sections based on the process of feature extraction. The researchers used a variety of feature extraction techniques and classification techniques, which are detailed here. The methodology for the pedestrian attribute recognition task is thoroughly explained in the following sections.

Chapter 3

Methodology

3.1 Introduction

The classification of an object in an image is known as image recognition. Images include spatial features that are used to classify the object. The use of conventional methods and deep learning methods for the pedestrian attribute recognition task is addressed in the last chapter. To conclude, convolutional neural networks outperformed conventional approaches significantly since CNN is good at capturing spatial features. The classification role is delegated to neural networks after the features have been captured. In the case of non-linearity, neural networks can perform exceptionally well.

Mask-RCNN will be used to capture pedestrians in a real-world scenario in this chapter. A quick description of how to choose the right CNN architecture for the pedestrian attribute recognition task will also be covered. Following that, the project's implementation will be demonstrated.

3.2 Overview of Proposed Pedestrian Attribute Recognition Framework

Figure 3.1 depicts the steps of the pedestrian attribute recognition system. A picture can contain multiple pedestrians in a real-world scenario. The Mask-RCNN object detector is used to isolate pedestrians from the input image in the first step of the pedestrian attribute recognition framework. The Mask-RCNN is used to extract isolated pedestrians during segmentation. After that, since the CNN model works on uniform image size, the image of the isolated pedestrian must be

resized. Scaling is applied to an image in order to minimise computational complexity. The image augmentation phase is in charge of introducing variance into the CNN model for training. It aids the CNN model in learning more effectively. After that, the CNN architecture uses the transformed image to extract spatial attributes. The classifier is a neural network that uses the extracted features as input and outputs a prediction of pedestrian attributes.

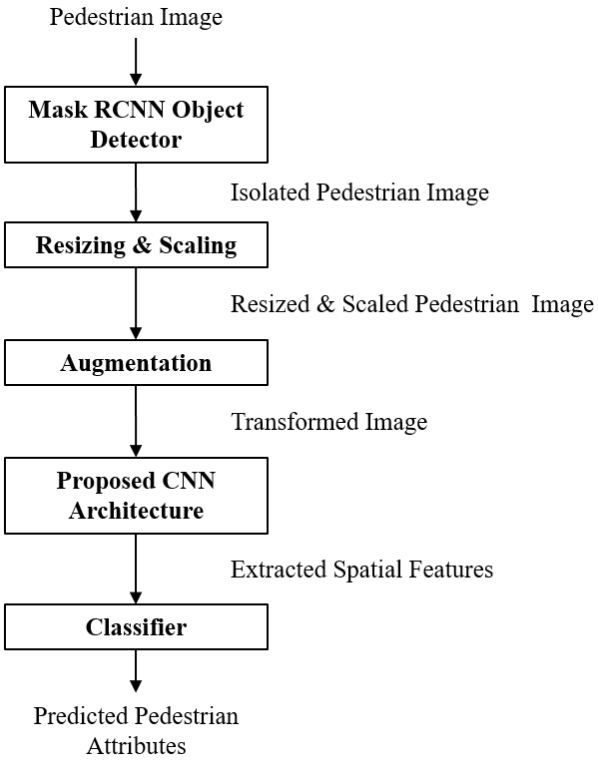


Figure 3.1: Steps of Proposed Pedestrian Attribute Recognition Framework.

The chosen CNN architecture phase is a series of experiments using the Transfer Learning method on a dataset with various CNN architectures. For the pedestrian attribute recognition task, a suitable CNN architecture is chosen after much experimentation.

3.3 Detailed Explanation

3.3.1 Mask-RCNN Object Detector

Pedestrian identification has advanced quickly in recent years, and it is now being used more widely. As described in [21], many applications depend on pedestrian detection, including smart vehicles, object tracking, robotics, and video surveillance. Pedestrian detection has gotten a lot of attention in the computer vision area because of its importance. Mask-RCNN or Mask Region-Based Convolutional Neural Network in [22] was developed as a result of a more in-depth investigation of deep learning for pedestrian detection. Figure 3.2 shows the basic overview of the Mask RCNN object detector framework.

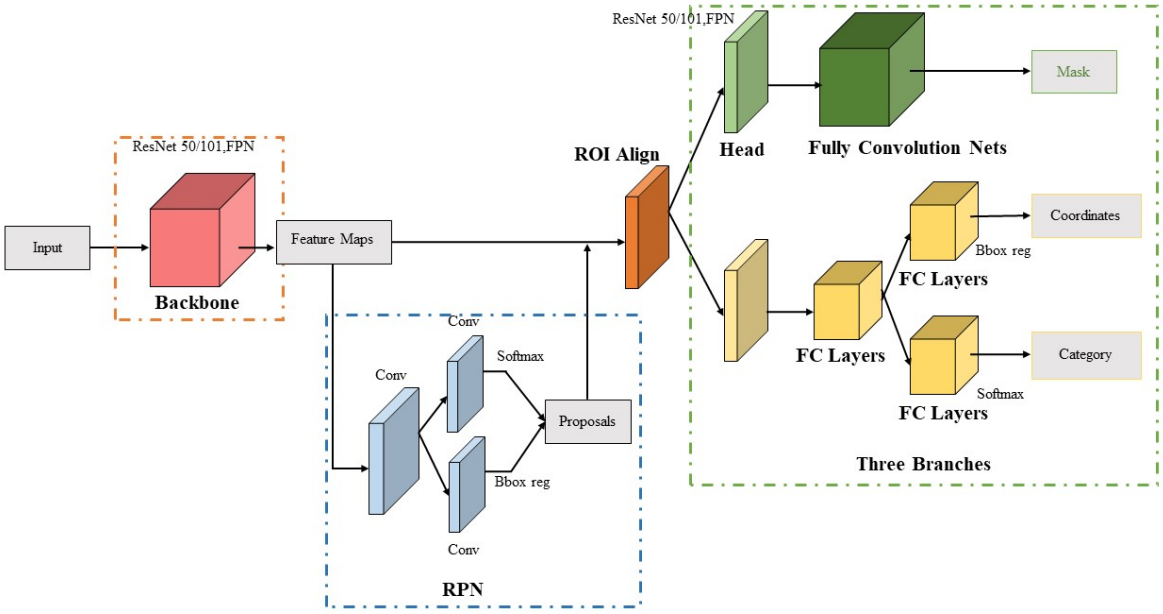
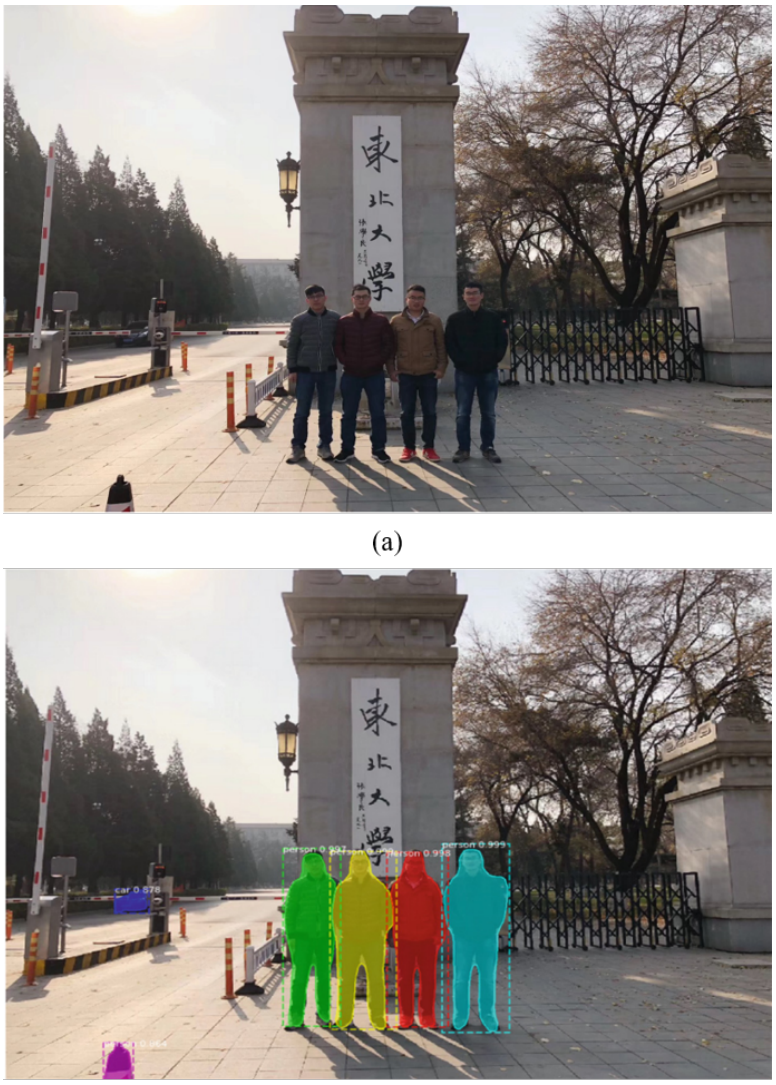


Figure 3.2: Mask RCNN Object Detector Framework.

The backbone on the Mask RCNN extracts features from the given input image. After that, the RPN proposed regions where objects are present in an image with the extracted features from the backbone. Then ROI Align Network works to give three types of output, i.e. the mask of the objects, the bounding box of the objects and the label or class-name of the objects. Figure 3.3 shows an example of Mask RCNN in real-world scenario in [23].



(a)

(b)

Figure 3.3: Demonstration of Mask RCNN Object Detector (a) Input Image (b) Output Image.

3.3.1.1 Backbone

ResNet 50, ResNet 101, or any Feature Pyramid Network can be used as the backbone of the Mask RCNN architecture (FPN). The feature maps of the input image are generated by the backbone network. This feature map will be used to suggest regions in a picture where an object is present.

3.3.1.2 Region Proposal Network

The Region Proposal Network (RPN) is a lightweight binary classifier that uses a CNN to generate multiple Regions of Interest (RoI). It accomplishes this by

placing anchor boxes on top of the picture. The classifier returns to object/no-object scores. Anchors with a high objectness score are subjected to non-max suppression.

3.3.1.3 Region of Interest Align Network

Instead of a single definite bounding box, the RoI Align network generates multiple bounding boxes that are warped into a fixed dimension. The warped features are then fed into completely connected layers for classification using softmax, and the regression model is used to refine the boundary box prediction. Warped features are also fed into the Mask classifier, which outputs a binary mask for each RoI using two CNNs. The Mask Classifier allows the network to create masks for all classes without causing competition.

3.3.2 Image Pre-Processing

Image processing is a technique for applying operations to an image in order to improve it or extract useful information from it. In Deep Learning, image pre-processing is needed to take advantage of the processed image. It has been demonstrated that image pre-processing improves the efficiency of a convolutional neural network significantly. Image pre-processing is done in our proposed framework using two techniques: resizing and scaling and augmentation. The following methods are listed in detail:

3.3.2.1 Resizing and Scaling

Image resizing means changing the resolution of an image. The main purpose of this task is to reduce computational complexity. As the convolutional neural networks take fix sized image as input, image resizing is necessary. Also, lower resolution means a smaller number of pixels in an image. Which will reduce computation cost. So taking these constraints, we resized all the RGB images into 150x150 resolution. The dataset has images of varying resolution from 33x81 to 415x583 as stated in [24]. Figure 3.4 shows an example of the resized image.

After the resizing, the images are scaled. The RGB images have pixel values of

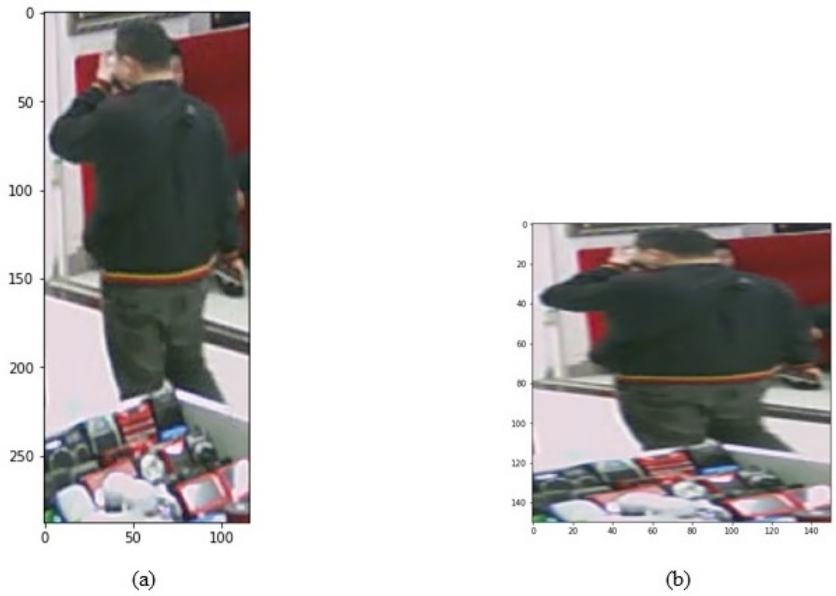


Figure 3.4: Resizing example: (a) Original Image (b) Resized Image.

range [0,255]. With a large pixel value, the computational cost also increases for the CNN. So to reduce the cost, all the pixel values are scaled in the range [0,1]. This is done by dividing the pixel values with 255 as 255 is the max pixel value in an RGB image.

3.3.2.2 Augmentation

Deep networks need a lot of training data. Image augmentation is typically needed to improve deep networks' performance to create a powerful image classifier with very little training data. Image augmentation artificially produces training images using various processing techniques or a mixture of methods, such as random rotation, turns, shear, and flips, among others.

When we train a CNN architecture with an image dataset, sometimes it causes the problem of overfitting. Overfitting occurs when the CNN architectures learn only on train data, and as a result, it doesn't perform well on test data. So if the variation of the image is introduced to the CNN model, then it performs better on test data. The variation is performed by the image augmentation technique.

The augmentation technique was applied to the processed image after resizing and scaling. We used augmentation during the training process in this study. There

are several batches of images in each epoch. Different augmentation techniques are used at random in each batch. We have, however, considered five different transformation methods. They are as follows:

- Width shift range 20
- Height shift range 20
- Rotate the images randomly by 25 degree
- Shear range 20
- Horizontal flip

Example of the augmentation techniques are shown in figure 3.5

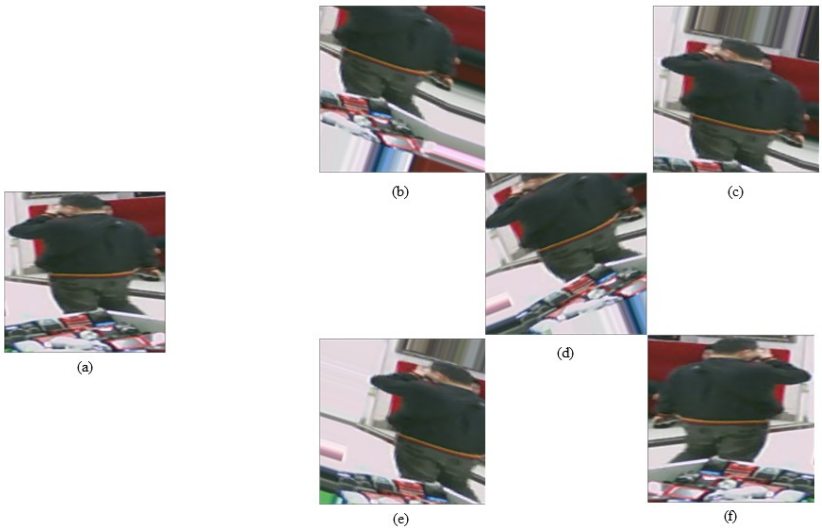


Figure 3.5: Augmentation example: (a) Original Image (b) Width Shift by 20% (c) Height Shift by 20% (d) Rotation by 25 Degree (e) Shearing by 20% (f) Horizontal Flip.

3.3.2.3 Normalization

Normalization is applied to standardise raw input pixels. It is a technique that transforms the input pixels of an image by subtracting the batch mean and then dividing the value by the batch standard deviation. The formula for standardisation is shown in equation 3.1.

$$z = \frac{x_i - \mu}{\sigma} \tag{3.1}$$

Here, x_i refers to input pixel, μ refers to batch mean, and σ refers to batch standard deviation.

When a pixel in a picture has a large size in comparison to others, it becomes dominant, and hence it causes the outcome of predictions to be less accurate.

Since the value of each pixel varies from 0 to 255, the scale of the image would be very large if we conduct an image classification operation. In this case, normalization is important as it makes a uniform distribution of the pixels as well as making the pixel values smaller, which makes computation faster. Also, normalization may cause convergence faster than unnormalized data.

3.3.3 Spatial Feature Extraction

An image has distinct characteristics. Features are the pixels in a picture that are responsible for the uniqueness of an entity in the image. Spatial feature extraction is the process of extracting features from an image that will later be used to describe or classify an entity. A convolutional neural network, which performs a series of operations, is used to extract the spatial features of an image. Convolution is followed by an activation function, which is followed by pooling. They are as follows:

3.3.3.1 Convolution Operation

The first layer of a CNN is the convolutional layer. It takes as input a matrix of dimensions [h1 * w1 * d1], which corresponds to the blue matrix in the 3.6 diagram. An image, or more precisely, a 3D Matrix, is used as the input.

- **KERNELS (filters):** A kernel is a matrix with the dimensions [h2 * w2 * d1], which in the figure 3.6 is one green matrix. The deep blue matrix in the figure 3.6, which has dimensions [h3 * w3 * d2], is the output for this layer. There are certain requirements that must be met:

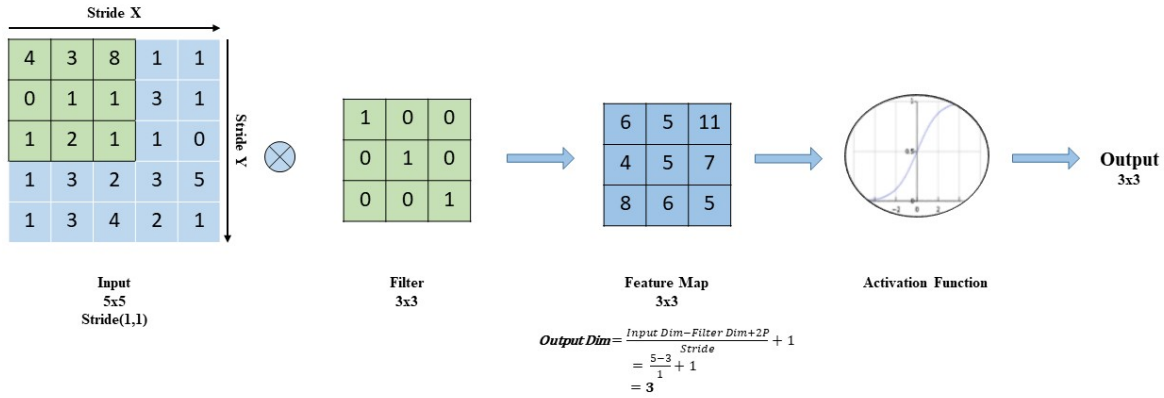


Figure 3.6: Working principle of Convolution Operation.

- 1) The input and kernel have the same depth (d1) (or number of channels).
- 2) The output's depth (d2) is equal to the value of kernels.

To calculate the dimension of the output matrix we can use equation 3.2:

$$Output\_Dimension = \frac{n - f + 2p}{s} + 1 \tag{3.2}$$

Where, n refers to the width/height of the input image.

f refers to the width/height of the kernel.

s refers to the width/height of the stride.

p is the padding value, which is 1 if the padding is the same; otherwise, it is 0.

The output or feature map is transferred to the activation function after the convolution process. For a CNN to study and understand something very difficult and non-linear dynamic functional mappings between the inputs and answer vector, activation functions are critical. The output signal would be a plain linear feature if we didn't have an Activation function. Our Neural network will be unable to learn and model other complex types of data, such as photographs and

videos, without the activation mechanism. When we map a Non-Linear equation, we see that it has a degree greater than one and that it has a curvature. To make the network more efficient and incorporate the capacity to learn something abstract and complicated from data and describe non-linear complex arbitrary functional mappings between inputs and outputs, we need to apply an activation function f(x). Figures 3.7 and 3.8 show examples of widely used activation functions.

- **Sigmoid activation function:** The function gives values in the range [0,1]. The higher the value of z, the closer the output goes to 1. Similarly, the lower the value of z, the closer the output goes to 0. It is mainly used in

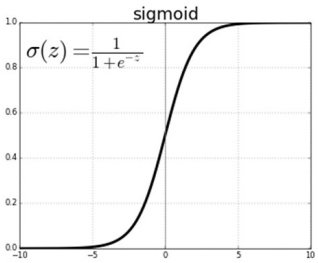


Figure 3.7: Sigmoid activation function.

the output layer for the binary classification task. The binary classification task means that the classifier will predict output in the range [0,1]. If the output is greater than 0.5, then it is considered as 1; otherwise, 0.

- **ReLU activation function:** Rectified Linear Unit is a function that gives values in the form max(0,z). This means that if the value of z is greater than 0, the output is z. Otherwise the output is 0.

  It is mainly used in the convolution layers as well as in the hidden layers of a neural network. It was recently demonstrated that it outperformed the Tanh activation function in terms of convergence by six times. It prevents and corrects the issue of vanishing gradients. ReLU is now used in almost all deep learning models.

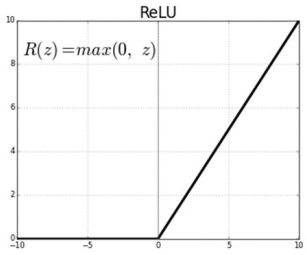


Figure 3.8: ReLU activation function.

3.3.3.2 Pooling Operation

Another important operation of CNN is the pooling operation. As the name indicates, it pools out the features from an image. Given an image, the pooling window takes the max value or the average value from it. So, two types of pooling operation can be performed.

- **Max Pool:** The following figure 3.9 describes the max pooling operation.

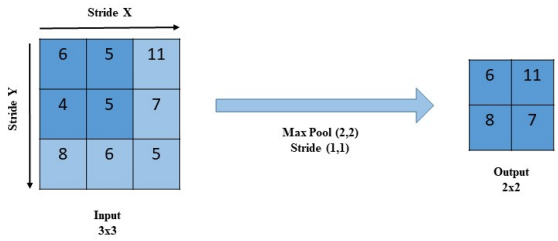


Figure 3.9: Working principle of Pooling Operation.

In the above figure, a window of 2x2 for the pooling operation is placed on the input. The maximum value of that window is chosen. This is the first

value in the output. With stride (1,1), the window is moved throughout the input. And finally, the output is produced.

- **Average Pool:** It is the same as max pool. The only difference is that instead of taking the maximum value from the window, the average value is taken.

3.3.4 Proposed CNN Architecture

As stated earlier, in our proposed framework, the CNN architecture is chosen after vivid experimentation. The experiment is performed on pre-trained convolutional neural networks. The transfer learning approach is chosen for the task. The details are given below:

3.3.4.1 Transfer Learning Approach

Deep Learning techniques require a large amount of data to learn. Sometimes this large amount is not available. Also, making a Deep Learning Network learn from scratch is very time-consuming. That's where the transfer learning technique comes. What the transfer learning technique does is that it transfers the knowledge from an already trained dataset. This knowledge is useful for detecting edges or other complex features. This knowledge can also be obtained by training a network from scratch, but this will only increase the cost. So to reduce time and cost, the transfer learning technique is chosen in our proposed architecture. There are many CNN architectures available for transfer learning purpose. These

Table 3.1: Top accuracy of CNN architectures on IMAGENET dataset.

Architecture Name	Top 5 Accuracy on IMAGENET Dataset	Top 1 Accuracy on IMAGENET Dataset
Inception ResNet V2	0.953	0.803
Xception	0.945	0.790
ResNet152V2	0.942	0.780
ResNet101V2	0.938	0.772

architectures are trained on a dataset. So we can call them pre-trained architectures. For our proposed framework, we have chosen four architectures based on their accuracy on the IMAGENET dataset in table 3.1. The data collected for the table 3.1 is in [25]. A side by side comparison among them is given below.

- **Transfer Learning with Inception ResNet V2 Architecture:**

  The Inception ResNet V2 architecture is pre-trained on the IMAGENET dataset as shown in table 3.1. It obtained 95.3% on top 5 accuracies. The IMAGENET dataset has variety in it. It has over 14M images and 1000 classes. The architecture can take an input image of dimension 299x299x3. But this large dimension of input will increase the computational cost. So we used 150x150x3 as the dimension of the input image. The architecture of Inception ResNet V2 is shown in figure 3.10 and 3.11.

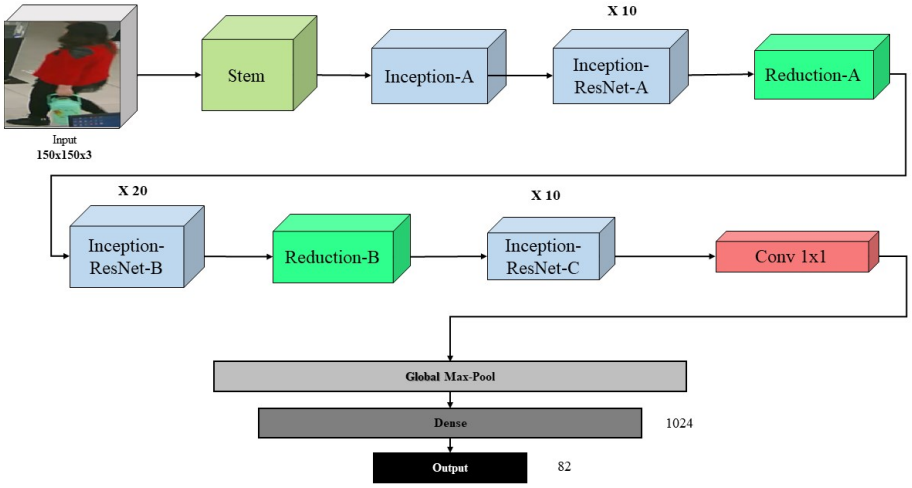


Figure 3.10: Overview of Inception ResNet V2.

The architecture consists of blocks where each block is a series of convolution and pooling operation. The network takes advantage of residual connection, which helps the network to learn from previous layers too. The inception concept is added in each residual block to reduce computation cost. After the last Conv 1x1 block, the extracted features are passed to the Global Max Pool layer. This layer performs the same operation as the Max pool, except the window size is the same as input width and height. So it reduces the no. of features as well as takes the most dominant feature in a channel. This layer also similar to Flatten layer as it also flattens the features. After much experimentation, a dense layer of 1024 nodes is chosen for the neural network to perform the classification task. And finally, an output layer of 82 nodes is chosen as our proposed architecture is a multi-label classification

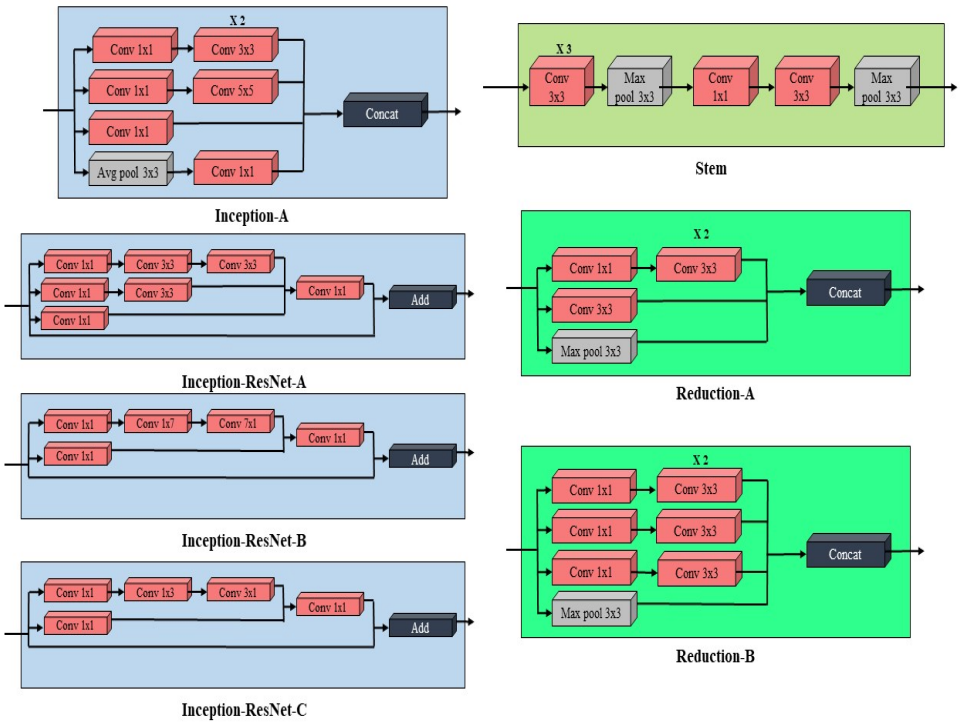


Figure 3.11: Details of the Blocks of Inception ResNet V2.

task with 54 binary attributes and 2 multi-label attributes, each multi-label attribute with 14 categories.

- **Transfer Learning with Xception Architecture:**

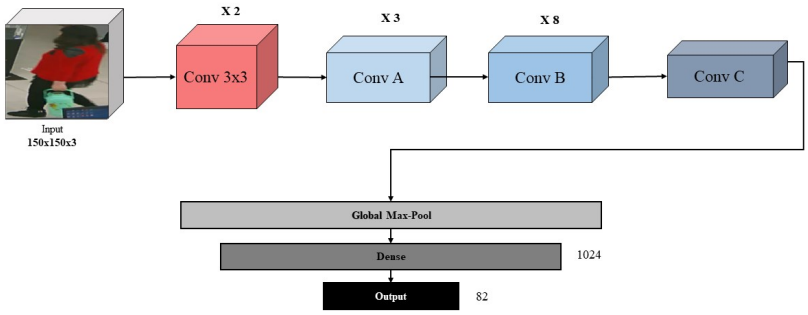


Figure 3.12: Overview of Xception.

The Xception architecture is pre-trained on the IMAGENET dataset as shown in table 3.1. It obtained 94.5% on top 5 accuracies. The architecture can take an input image of dimension 299x299x3. But as the computational

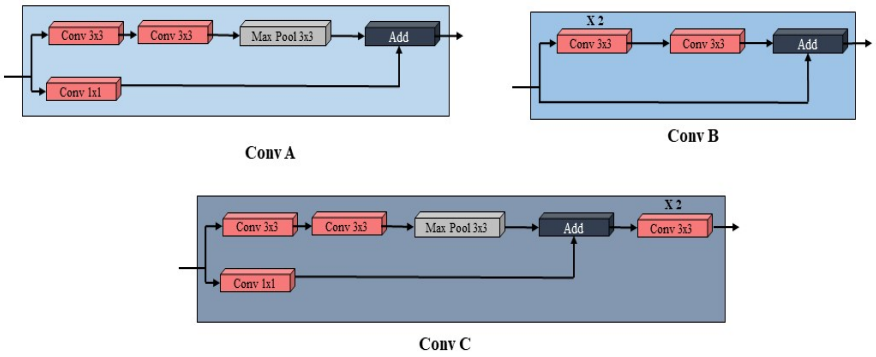


Figure 3.13: Details of the Blocks of Xception.

cost will rise, we used 150x150x3 as the dimension of the input image. The architecture of Xception is shown in figure 3.12 and 3.13.

The architecture consists of blocks where each block is a series of convolution and pooling operation. The network also takes advantage of residual connection, which helps the network to learn from previous layers too. The inception concept is added in each residual block to reduce computation cost. After the Conv C block, the rest of the network is the same as the Inception ResNet V2 network. All the networks have the same Global Max Pool layer, Dense layer and Output layer to make consistency.

- **Transfer Learning with ResNet 152 V2 Architecture:**

  The ResNet 152 V2 architecture is pre-trained on the IMAGENET dataset as shown in table 3.1. It obtained 94.2% on top 5 accuracies. The architecture can take an input image of dimension 224x224x3. But as the computational cost will rise, we used 150x150x3 as the dimension of the input image. The architecture of ResNet 152 V2 is shown in figure 3.14. The architecture consists of blocks where each block is a series of convolu-

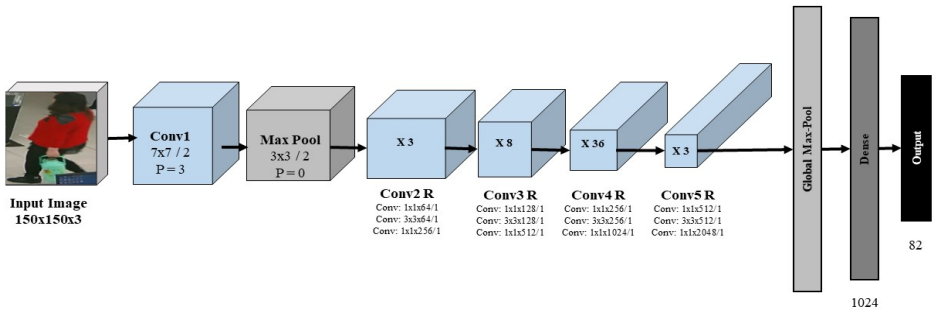


Figure 3.14: Overview of ResNet 152 V2.

tion and pooling operation. The network also takes advantage of residual connection, which helps the network to learn from previous layers too. The inception concept is added in each residual block to reduce computation cost. After the Conv5 R block, where R indicates a residual connection in the block, the rest of the network is the same as the Inception ResNet V2 network. All the networks have the same Global Max Pool layer, Dense layer and Output layer to make consistency. This architecture performed better than the others. The details are discussed in the next chapter.

- **Transfer Learning with ResNet 101 V2 Architecture:**

  The ResNet 101 V2 architecture is the same as ResNet 152 V2 3.1. It obtained 93.8% on top 5 accuracies. The architecture can take an input image of dimension 224x224x3. But as the computational cost will rise, we used 150x150x3 as the dimension of the input image. The architecture of ResNet 101 V2 is shown in figure 3.15. The only difference between ResNet 152 V2 and ResNet 151 V2 is the blocks are repeated fewer number times in ResNet 101 V2. All the networks have the same Global Max Pool layer, Dense layer and Output layer to make consistency.

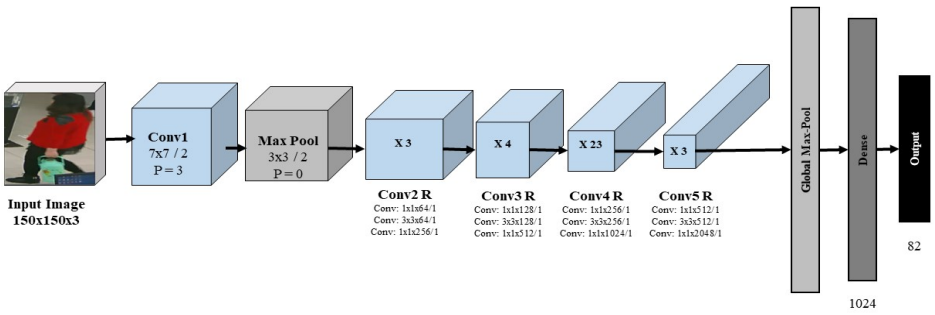


Figure 3.15: Overview of ResNet 101 V2.

3.3.5 Classifier

After the spatial feature extraction from the CNN architecture, a classifier is used to predict the pedestrian attributes. As a neural network is used in our proposed architecture, the classifier has one dense layer. Dense layers are required to give efficiency to a neural network. Because as the problem becomes complex, the network needs to learn complex non-linearity. The dense layer introduces non-linearity as the ReLU activation function is used in every node. In our case, we used 1024 neurons in the dense layer. After the dense layer, the output layer is added. It is the layer that helps classify an object. As our proposed architecture is a multi-label classification task meaning multiple attributes can be predicted for a pedestrian, the activation function is changed here. We used the sigmoid activation function as it gives a probability between 0-1. If the probability is greater than 0.5, we considered the attribute positive, meaning the attribute is present in the pedestrian. Otherwise, we considered it as 0, which means absent. We can get the output from an output layer by the equation 3.3

$$Z = f(W.X + b) \tag{3.3}$$

Where W and X are weight matrix and input matrix, respectively. b is the bias of every node of the input layer. f represents the activation function which is the sigmoid activation function in our case. Z is the output. The sigmoid activation

function is shown in figure 3.7.

3.4 Implementation Details

3.4.1 Dataset and Labels

The Richly Annotated Pedestrian (RAP) v2 dataset was selected for our research because it acts as a single benchmark for both attribute-based and image-based individual retrieval in actual surveillance scenarios. RAP v2 is a large-scale dataset that includes 84,928 photographs of 72 different categories of attributes, as well as tags for perspective, occlusion, body sections, and 2,589 different human identities.

- The dataset is split into train, validation and test data. The ratio for the split is 6:2:2 (50862:16951:16953) accordingly.
- 54 Binary Attributes and 2 Multi-label Attributes are chosen as shown in table 3.2 for pedestrian attribute recognition task.

Table 3.2: Pedestrian Attributes Used in PAR framework.

Pedestrain Attributes	
Binary Attributes	Femal, AgeLess16, Age17-30, Age31-45, Age46-60, BodyFat, BodyNormal, BodyThin, Customer, Employee, hs-BaldHead, hs-LongHair, hs-BlackHair, hs-Hat, hs-Glasses, ub-Shirt, ub-Sweater, ub-Vest, ub-TShirt, ub-Cotton, ub-Jacket, ub-SuitUp, ub-Tight, ub-ShortSleeve, ub-Others, lb-LongTrousers, lb-Skirt, lb-ShortSkirt, lb-Dress, lb-Jeans,lb-TightTrousers, shoes-Leather, shoes-Sports, shoes-Boots, shoes-Cloth, shoes-Casual,shoes-Other, attachment-Backpack, attachment-ShoulderBag, attachment-HandBag, attachment-Box, attachment-PlasticBag, attachment-PaperBag, attachment-HandTrunk, attachment-Other, action-Calling, action-Talking, action-Gathering, action-Holding,action-Pushing, action-Pulling, action-CarryingByArm, action-CarryingByHand, action-Other
Multi-label attributes	ub-ColorBlack, ub-ColorWhite, ub-ColorGray, up-ColorRed, ub-ColorGreen, ub-ColorBlue, ub-ColorSilver, ub-ColorYellow, ub-ColorBrown, ub-ColorPurple, ub-ColorPink, ub-ColorOrange, ub-ColorMixture, ub-ColorOther
	lb-ColorBlack, lb-ColorWhite, lb-ColorGray, lb-ColorRed, lb-ColorGreen, lb-ColorBlue, lb-ColorSilver, lb-ColorYellow, lb-ColorBrown, lb-ColorPurple, lb-ColorPink, lb-ColorOrange, lb-ColorMixture, lb-ColorOther

3.4.2 Image Pre-processing

As discussed in section 3.3.2, image pre-processing is performed for our proposed framework. Resizing, scaling as well as augmentation techniques are used for the

pre-processing. Also, an experiment is also conducted on normalized and unnormalized data to check the necessity of normalization in our proposed framework.

3.4.3 Freezing Layers

For the transfer learning purpose, every layer except the last 10 layers of backbone architecture is frozen. As the first few layers have already knowledge of edges, we kept the knowledge of these layers by freezing them. As the last few layers are responsible for detecting important features of images, we trained those layers.

3.4.4 Hyperparameter Setting

A Convolutional Neural Network's hyperparameters are those that the model can't discover on its own. These variables are in charge of getting the learning to the convergence stage. This criteria can only be determined by trial and error. After a lot of trial and error, we came up with the following hyperparameters for our proposed architecture:

- **Initial learning rate:** 0.01. After every epoch the learning rate will change by checking some conditions. The factors are:
  - **Factor** = 0.5; i.e. learning_rate = learning_rate*0.5
  - **Patience** = 3; number of epochs with no improvement after which training will be stopped.
  - **Min_lr**=$1e^{-8}$; This is the minimum learning rate.
  - **Min_delta**=$1e^{-4}$; minimum change in the monitored quantity to qualify as an improvement.
- **Batch size:** 64
- **Epochs:** 40
- **Optimization algorithm:** Adam optimizer is used as it gives better performance than other optimizers in most cases.
- **Dense layer with nodes:** 1024 nodes

- **Output layer with nodes:** 82 nodes or 54 nodes based on the experiments
- **Loss function:** binary cross-entropy' loss is used here as this is a binary classification task. The formula for the binary cross-entropy loss can be calculated with the formula 3.4.

$$-\frac{1}{N}\sum_{i=1}^{N} y_i.log(p(y_i)) + (1-y_i).log(1-p(y_i)) \tag{3.4}$$

  Here, y_i represents the actual class and $log(p(y_i)$ is the probability of that class.

3.4.5 Hardware Configuration

The trial was carried out on Google Colab, a freely accessible open cloud service operated by Google. A total of 12 GB RAM was given, with 12-hour runtime. There was also 14 GB of Tesla K80 GPU memory available. The CNN framework was built using Keras with Tensorflow 2.0 as the backend and Python 3.7 as the programming language.

With these configurations, the architectures are trained. After that, the best performing architecture is chosen for the pedestrian attribute recognition task.

3.5 Conclusion

This chapter goes into the technique for Pedestrian Attribute Recognition. By using a transfer learning technique, the suggested method has been tested for various CNN architectures. This experiment yielded a CNN architecture. The experimental result analysis of the proposed methodology is discussed in the following chapter.

Chapter 4

Results and Discussions

4.1 Introduction

In the previous chapter, a detailed description of the methodology for Pedestrian Attribute Recognition was discussed. This chapter examines the performance of the proposed method briefly.

As a dataset is available for this task, the performance is measured with this dataset. This chapter also includes the comparison among CNN architectures.

4.2 Dataset Description

Many datasets are available for the purpose of recognizing pedestrian attributes. D.Li et al. proposed the Richly Annotated Pedestrian (RAP) dataset with a total of 41,585 pedestrian samples annotated with 72 attributes from real multi-camera surveillance scenarios in [8]. Later, in [24], D. Li et al. proposed the RAP v2 dataset, which is a large-scale dataset that comprises of 84,928 images with 72 categories of attributes and additional identifiers.

The resolution of the images in the dataset varies from 33 $\times$ 81 to 415 $\times$ 583. 25 camera scenes from an indoor shopping mall captured the images. We used the ratio of (6: 2: 2) as stated in the last chapter. Here 60% data is used for training and 20% for testing and validation. The dataset is highly imbalanced. For consistency, the split for the dataset taken is the same for other previous work.

54 binary attributes and 2 multi-label attributes are selected for the PAR framework. As the dataset is imbalanced the distribution of positive samples for the dataset is given below:

- **Distribution of Data for 82 Attributes**:
  - **Distribution of Train Data:** The distribution of the positive samples for the train data is shown in figure 4.1.

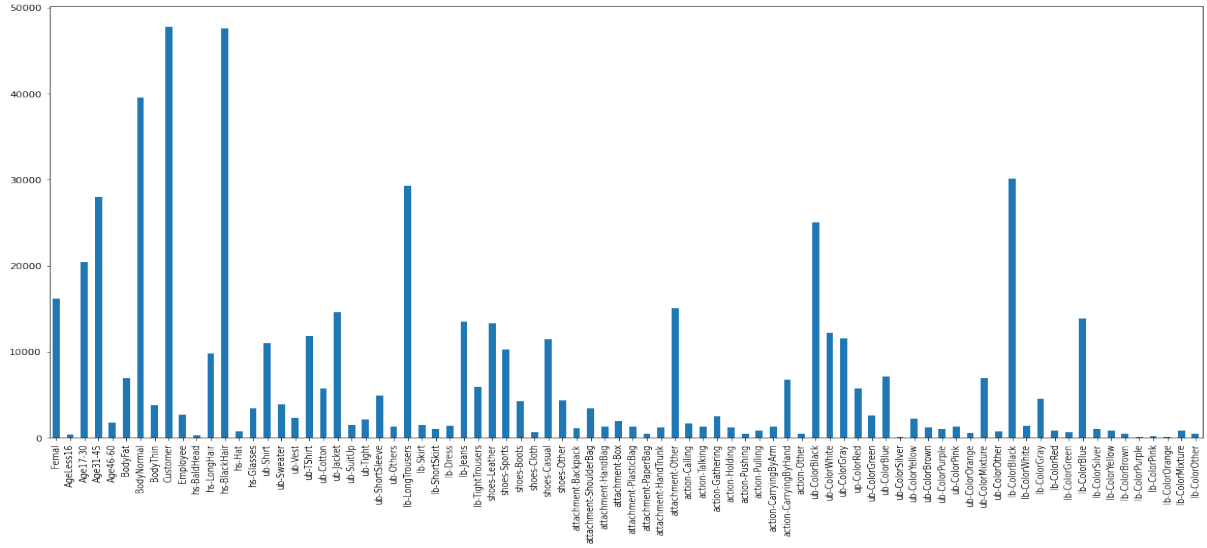


Figure 4.1: Distribution of Positive Samples for Train Data with 82 Attributes.

  - **Distribution of Test Data:** The distribution of the positive samples for the test data is shown in figure 4.2.

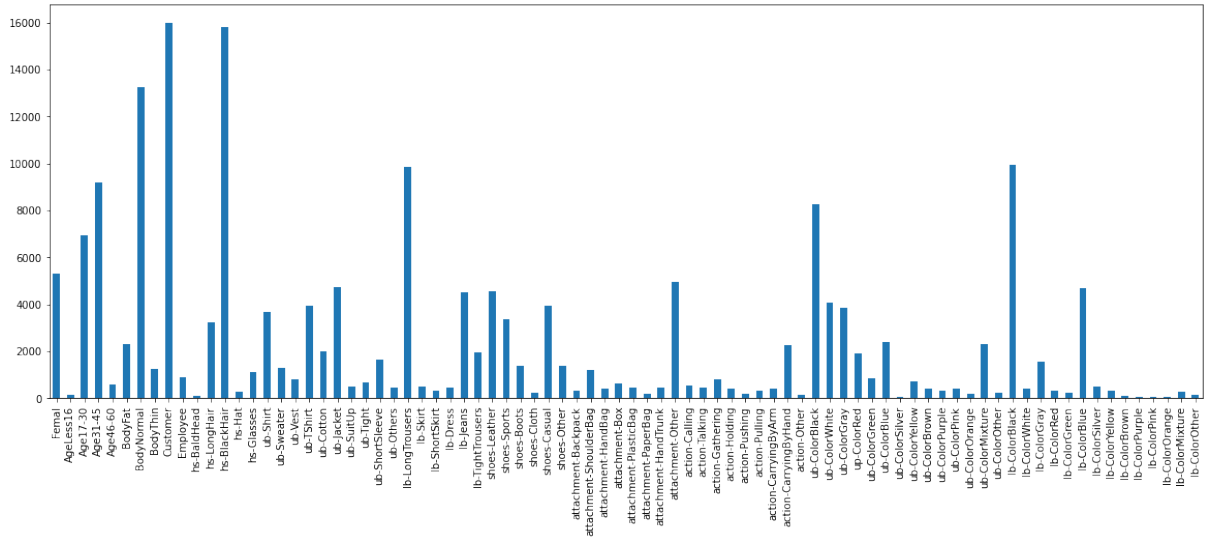


Figure 4.2: Distribution of Positive Samples for Test Data with 82 Attributes.

– **Distribution of Validation Data:** The distribution of the positive samples for the validation data is shown in figure 4.3.

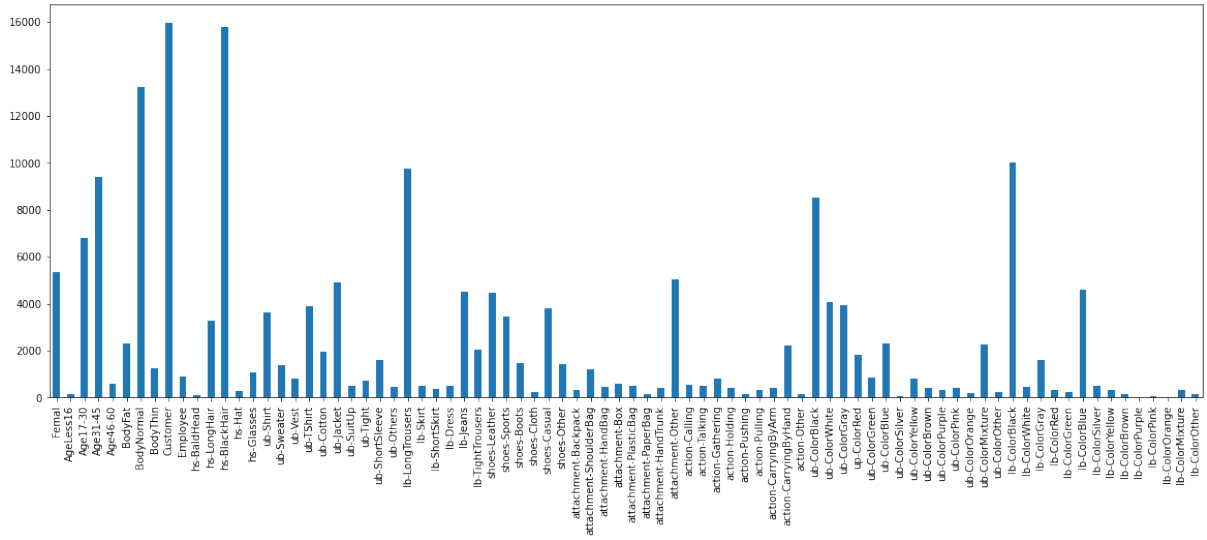


Figure 4.3: Distribution of Positive Samples for Validation Data with 82 Attributes.

- **Distribution of Data for 54 Attributes**: The 54 attributes are shown in table 3.2. The binary attributes row in the table are the 54 attributes considered in our proposed architecture and also in other existing works.

  – **Distribution of Train Data:** The distribution of the positive samples for the train data is shown in figure 4.4.

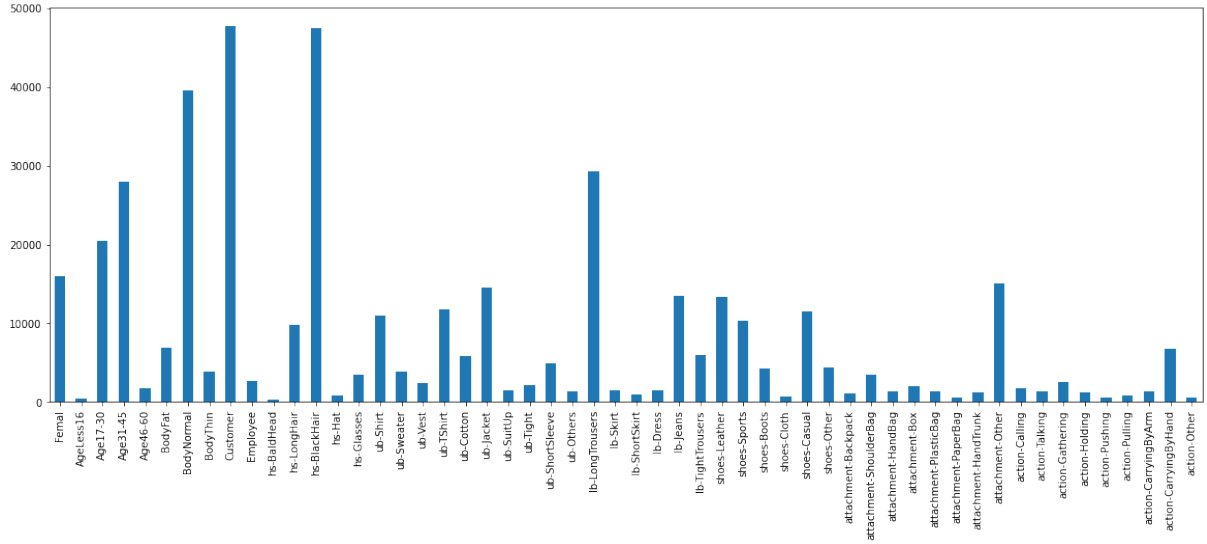


Figure 4.4: Distribution of Positive Samples for Train Data with 54 Attributes.

– **Distribution of Test Data:** The distribution of the positive samples for the test data is shown in figure 4.5.

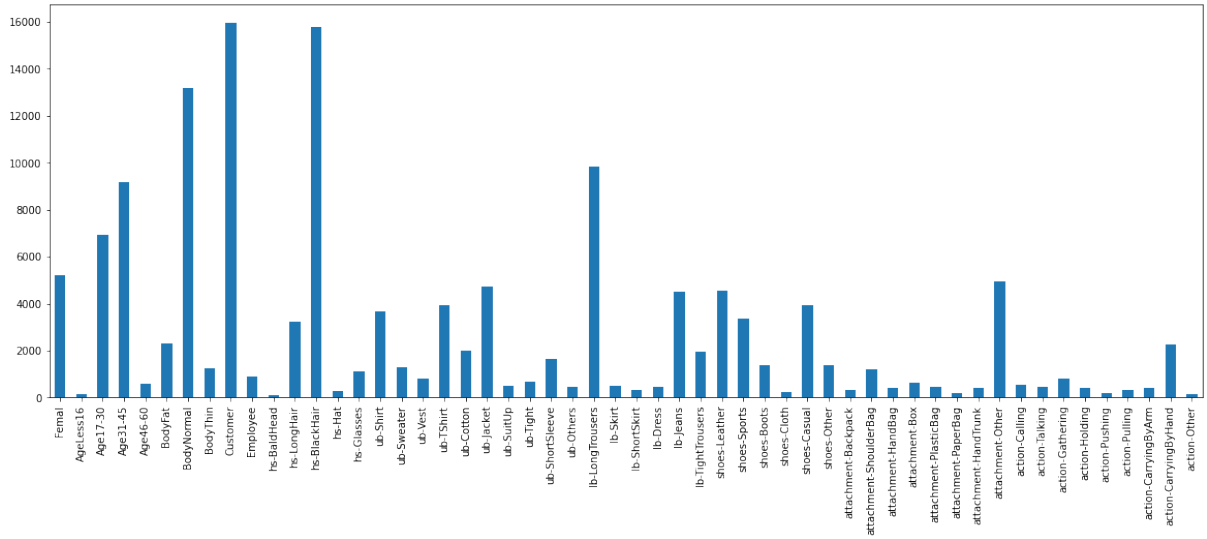


Figure 4.5: Distribution of Positive Samples for Test Data with 54 Attributes.

– **Distribution of Validation Data:** The distribution of the positive samples for the validation data is shown in figure 4.6.

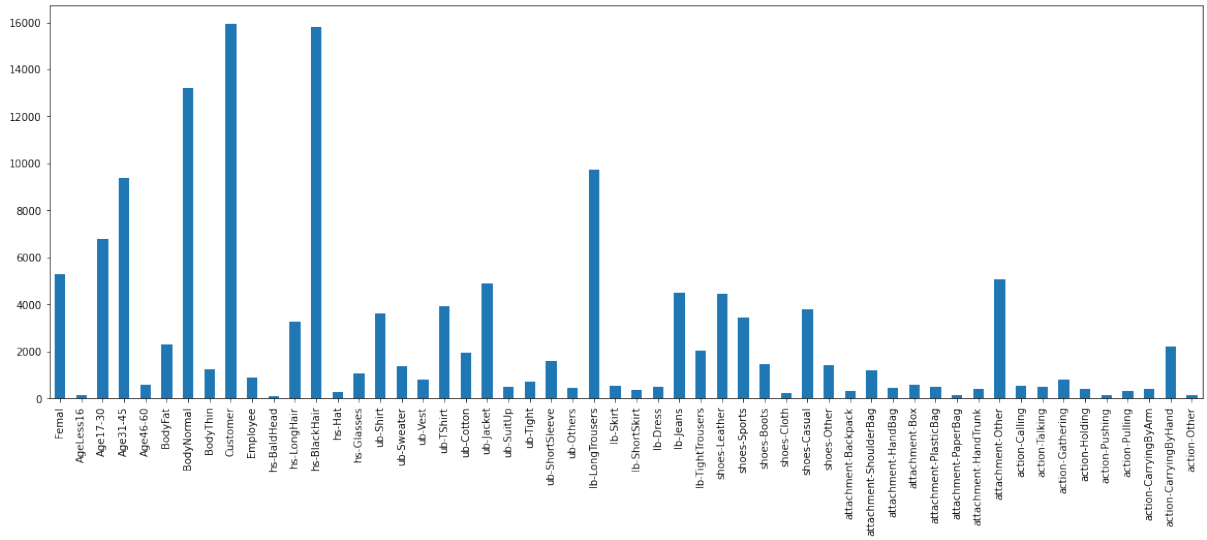


Figure 4.6: Distribution of Positive Samples for Validation Data with 54 Attributes.

Example of the RAP v2 dataset is shown in figure 4.7.

Figure 4.7: Example of RAP v2 Dataset.

4.3 Evaluation Metrics

In order to assess our proposed CNN architecture, we used some metrics. Previous methods also use these metric for evaluating their work. Accuracy, Precision, Recall, F1 score are the most used evaluation metrics for the pedestrian attribute recognition task.

These metrics help to decide which CNN architecture is more efficient and gives a better result. Multiple evaluation metrics are necessary as we can not determine the efficiency and performance of an algorithm by a single metric. As our task is the multi-label classification, we have also used a confusion matrix to visualize per attribute result. They are discussed briefly in this section.

4.3.1 Confusion Matrix

The Confusion Matrix is an efficiency metric for classification problems including machine learning and deep learning, where the result may be two or more groups. There are four distinct variations of expected and actual values in this table. The confusion matrix is shown in figure 4.8 It's great for determining Recall, Precision, Specificity, Accuracy, and, most notably, the AUC-ROC Curve. The terms used in figure 4.8 of the confusion matrix is described below:

- **True Positive:** The model predicted positive, and the actual label is also positive
- **True Negative:** The model predicted negative, and the actual label is also negative

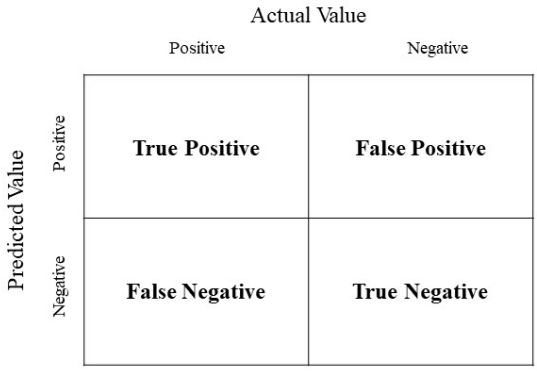


Figure 4.8: Confusion Matrix.

- **False Positive:** The model predicted positive, but the actual label is negative
- **False Negative:** The model predicted negative, but the actual label is positive

The other evaluation metrics that are used in our proposed architecture are discussed briefly below:

- **Accuracy:** Accuracy means how many of each class did the model correctly predict. This considers both TP (True Positive) and TN (True Negative). Accuracy can be measured by the following formula in 4.1

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{4.1}$$

- **Precision:** Precision means how many of the positive groups expected by the model are actually positive. Precision can be measured by the following formula in 4.2

$$Precision = \frac{TP}{TP + FP} \tag{4.2}$$

- **Recall:** Recall means how much of each positive class did the model correctly predict. Recall can be measured by the following formula in 4.3

$$Recall = \frac{TP}{TP + FN} \tag{4.3}$$

- **F1 score:** It's difficult to compare two models that have low precision but high recall, or the other way around. But we use the F1 score to compare them. An F1 score is a tool for measuring both recall and precision at the same time. It employs a Harmonic Mean instead of Arithmetic Mean, punishing extreme values more severely. F1 score can be measured by the following formula in 4.4

$$F1Score = \frac{2 * Precision * Recall}{Precision + Recall} \tag{4.4}$$

4.4 Impact Analysis

Our proposed work can have a great impact on society and the environment as well as on ethics. As we recognise pedestrian attributes by means of computer vision, this has a huge impact on society. They are briefly discussed in this section.

4.4.1 Social and Environmental Impact

A pedestrian attribute recognition framework can recognise attributes from a pedestrian image. With this attribute information, a person can be re-identified from any surveillance scenario. So we can track a person from any place just with pedestrian attributes. It has both merits and demerits.

- **Merits:**
  - The government can identify criminals with the help of PAR framework
  - Any organisation can use this framework for security purposes.
  - Crime rate will reduce.
- **Demerits:**

- If this framework is used by the wrong people, then this could cause chaos.
- People will have no privacy as they will be monitored.

4.4.2 Ethical Impact

Our proposed framework has an ethical impact too. As this is a powerful work, only the government has the power to implement it in surveillance scenario. So, the government is held responsible for this type of work. The ethical issues are if this work is used in the wrong, it will hamper public privacy as we can track a person by the attribute information. Moral obligation before using this work is mandatory.

4.5 Evaluation of Framework

In our proposed framework, we chose 54 binary attributes and 2 multi-label attributes for recognising pedestrian attributes. As our approach is transfer learning-based, as stated in the last chapter, initially, some CNN architectures are selected based on their performance on the IMAGENET dataset as described in table 3.1. The step by step evaluation of our framework is described in this section.

4.5.1 Comparison of Pre-trained Models

The overview of the architectures is described in section 3.3.4.1. We performed experiments on these architectures. The number of epochs for the experiments is 40. From table 4.1 we can see that ResNet 152 V2 performed better than other architectures. It obtained 92.14% accuracy on the RAP v2 dataset. So, initially, we chose ResNet 152 V2 CNN architecture for our proposed framework.

4.5.2 Experiment on Normalized Data

As stated previously, we have experimented on normalized and unnormalized data to see the difference in performance. The performance comparison of the

Table 4.1: Comparison of Pre-trained Models on RAP v2 Dataset.

Pre-trained Models	**Trainable Parameters**	**Nodes in Dense Layer**	**Test Accuracy**(%)
Inception ResNet V2	1.6M	1024	91.61
Xception	2.1M		91.56
ResNet152V2	2.1M		**92.14**
ResNet101V2	2.1M		92.11

ResNet 152 V2 architecture on normalized and unnormalized data are represented in Table 4.2.

Table 4.2: Effect of Normalization on Performance.

	Training Accuracy(%)	**Validation Accuracy**(%)	**Test Accuracy**(%)	**Total Epochs**
Normalized Data	92.45	92.48	92.45	40
Unnormalized Data	94.25	93.46	93.41	40

The table shows that our normalized data didn't perform better than unnormalized data. The reason behind it is that we have already scaled our image data in the range [0,1]. So the task of normalization is already is done by scaling. That is why normalization on the image data didn't have much effect. Also, scaling made the convergence faster in our proposed architecture as we used a larger learning rate.

4.5.3 Imbalanced Dataset Experiment on ResNet 152 V2 with 82 Attributes

After selecting primarily ResNet 152 V2, we performed vivid experiment on this architecture. For using the transfer learning method, we tried to freeze layers of the architecture. Here, freezing means the weights of the frozen layers will remain unchanged in the training phase. No data balancing techniques are applied in this experiment. The purpose is to choose an architecture for performing a further experiment. After much experimentation table 4.3 was obtained.

From the table 4.3, we can see that the accuracy is increased by freezing the last

Table 4.3: Experiment on ResNet 152 V2 Via Transfer Learning Method.

Experiment	**ResNet 152 V2 Version**	**Trainable Layers (Excluding FC Layer)**	**Nodes in Dense Layer**	**Trainable Parameters**	**Test Accuracy** (%)	**Test F1 Score** (%)
Experiment 1	ResNet 152 V2	None	1024	2,184,274	92.04	27.58
Experiment 2	ResNet 152 V2	All		60,372,175	93.16	36.70
Experiment 3	ResNet 152 V2	Last 4 layers		3,238,994	91.96	24.61
Experiment 4	ResNet 152 V2	Last 8 layers		5,599,314	92.61	34.08
Experiment 5	ResNet 152 V2	Last 12 layers		6,648,914	93.09	40.16
Experiment 6	ResNet 152 V2	Last 14 layers		6,653,010	**93.41**	**45.18**

14 layers of ResNet 152 V2 architecture. For consistency, we used a dense layer of 1024 nodes in all cases and 40 epochs per experiment. 93.41% accuracy is obtained by the architecture.

The graphical comparison among the experiments can be observed in figure 4.9.

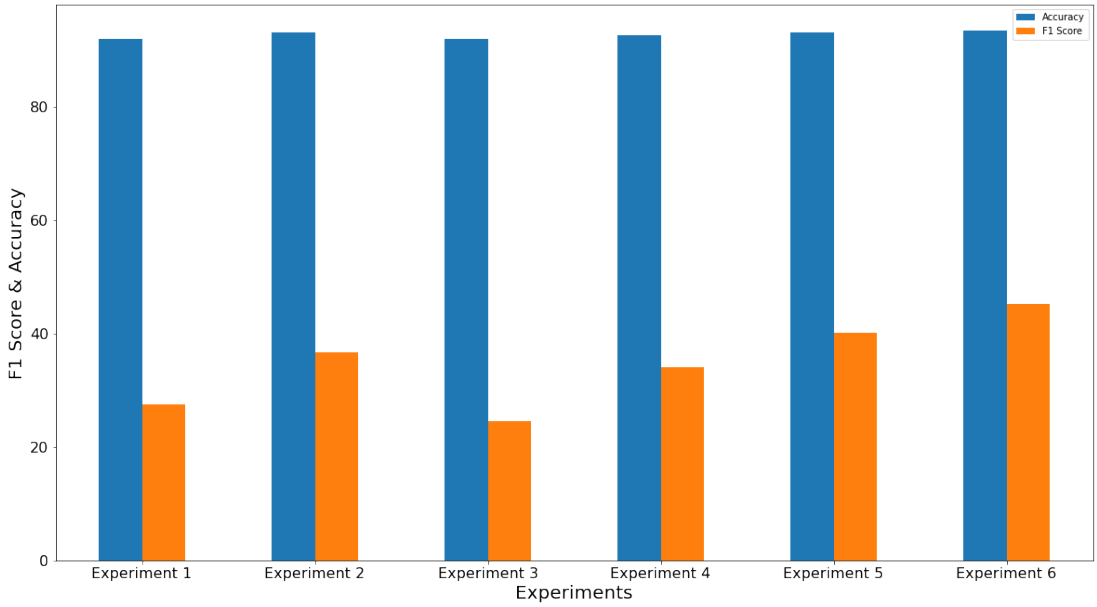


Figure 4.9: Accuracy F1 score of Experiments on Primary Architecture.

4.5.3.1 Loss and Accuracy Curve of ResNet 152 V2

As we obtained ResNet 152 v2 as the better performer among other CNN architectures, the training and validation curve is also obtained to check if our chosen CNN architecture is overfitted on the training data or not. In our architecture, we have chosen the Hard sharing of features method. Hard sharing means sharing the features for all our selected attributes in the Fully Connected layers. In our model, the Global Max pool, Dense and Output layers are considered Fully Connected Layer.

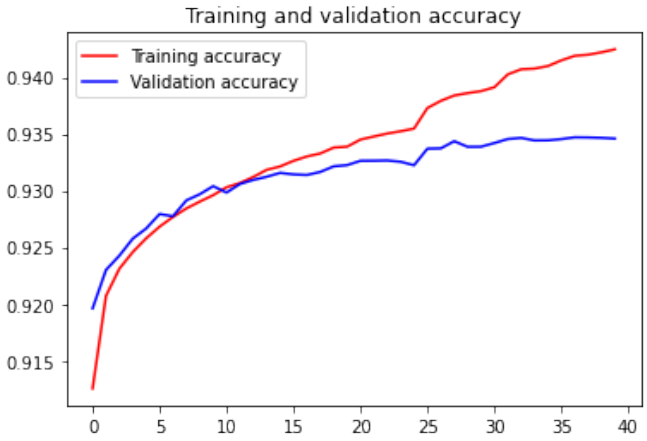


Figure 4.10: Accuracy of ResNet 152 V2

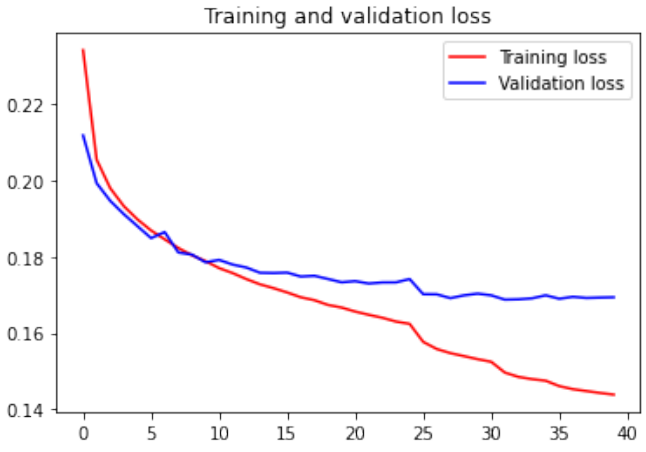


Figure 4.11: Loss of ResNet 152 V2.

The purpose of choosing hard sharing features is that attributes have interdependency. For example, gender attribute is related to shirt, pant, skirt attribute. Because if gender is male, then the person will wear a shirt, pant otherwise skirt. So taking these things into account, we chose hard sharing of features techniques.

As a result, our model learned efficiently. The loss and accuracy curve is shown in figure 4.11 and figure 4.10.

From figure 4.11 and 4.10 we can deduce that our model learned efficiently. The model is neither overfitted nor underfitted on train data.

4.5.3.2 Confusion Matrix of ResNet 152 V2 Architecture

Pedestrian attribute recognition is a multi-label classification task. As for multi-

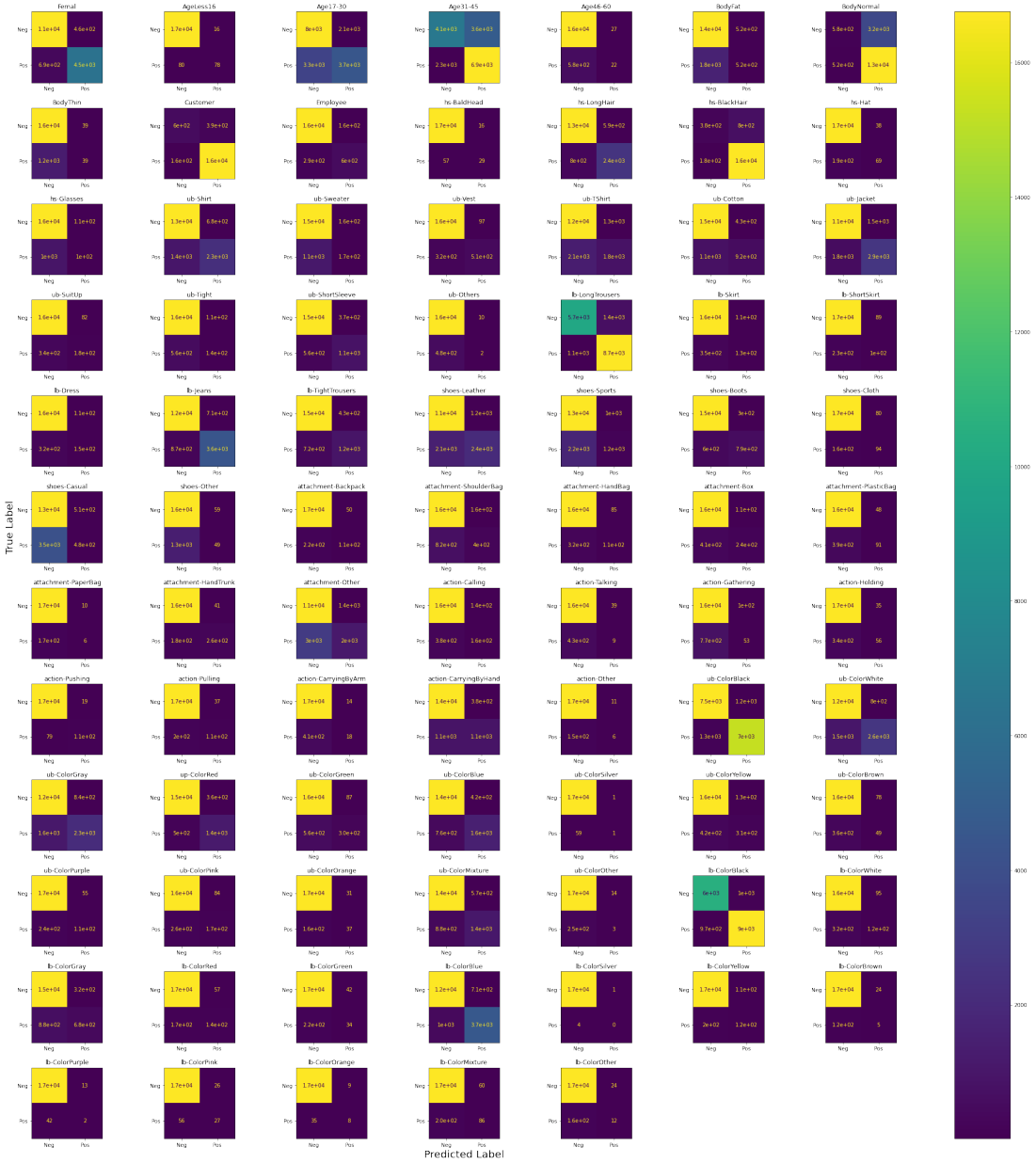


Figure 4.12: Confusion Matrix of ResNet 152 V2.

label classification, a single confusion matrix can not describe the result. So for

every label, a confusion matrix is required. Keeping this in mind, we showed the confusion matrix for our proposed architecture in figure 4.12.

Figure 4.12 shows the confusion matrices used by this network to evaluate results. It can be shown that certain classes have been misclassified due to semantic similarity with the misclassified classes.

4.5.4 Balanced Dataset Experiment on ResNet 152 V2 with 82 Attributes

As our dataset RAP v2[24] is highly imbalanced, we also performed some data balancing experiments to compare the performance of our proposed architecture in both scenarios. For the multi-label classification task, we choose the over-sampling technique for balancing our data.

- **Oversampling**: It is a technique that finds examples in the minority class and duplicates them. For multi-label classification, it is a bit different. The process to perform oversampling in our proposed framework is given below:
  - Give every data a power value
  - Find the frequency of power values
  - Take the max power value
  - For each power value :
    - Get size of data = (max power value – power value)
    - Take a random copy from the dataset with the same power value
    - Add it with the amount of size of data in the original dataset
- **Weighted binary cross-entropy loss**: This loss function is similar to binary cross-entropy loss function as stated in 3.4. The weights of the positive and negative samples are integrated in this loss function to make the architecture learn minority samples. The weights are calculated by the following steps:
  - For each attribute:

* Positive weight, w_p = Total data size/ 2 x Total Positive Samples

* Negative weight, w_n = Total data size/ 2 x Total Negative Samples

– Apply the weights to the following loss function in 4.5:

$$-\frac{1}{N}\sum_{i=1}^{N} w_p * y_i.log(p(y_i)) + w_n * (1 - y_i).log(1 - p(y_i)) \quad (4.5)$$

Figure 4.13 shows the distribution of power values before applying the oversampling technique.

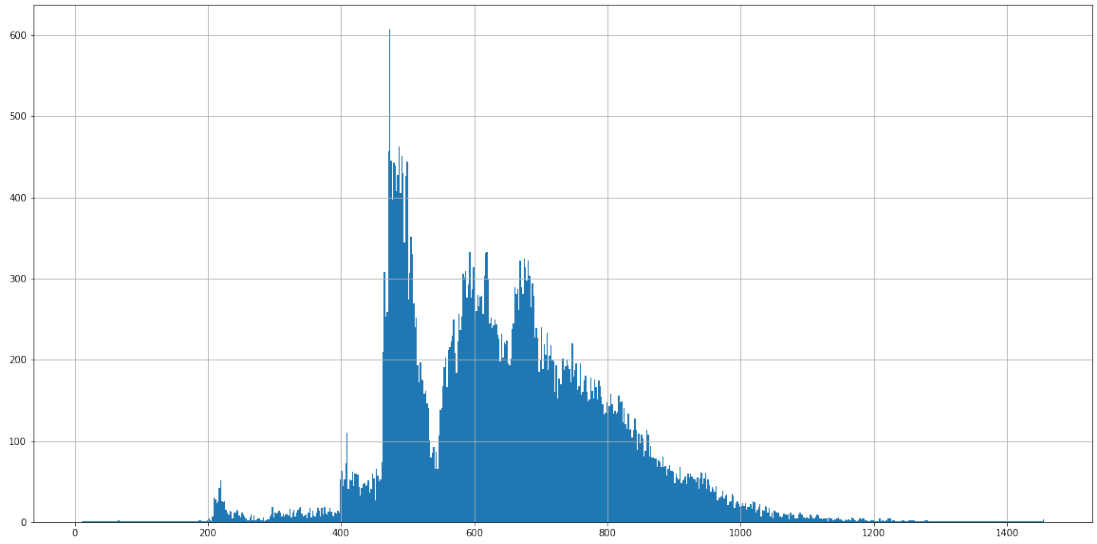


Figure 4.13: Distribution of Power Values Before Oversampling.

After applying the oversampling technique, we get the figure 4.14. The power values are now equally distributed. As this is a histogram, the x-axis represents the power values and the y-axis represents the frequency of power values.

Finally, the positive label distribution of the training data is obtained in figure 4.15. The training data has increased significantly because of applying the oversampling technique. The total number of train images are 330207.

4.5.4.1 Loss and Accuracy Curve of ResNet 152 V2

As ResNet 152 v2 has performed better than other architectures, we conducted data balancing experiment with same hyperparameters with this architecture. Applying the oversampling and weighted binary cross-entropy loss, we get the loss and accuracy curve in figure 4.16 and 4.17 respectively.

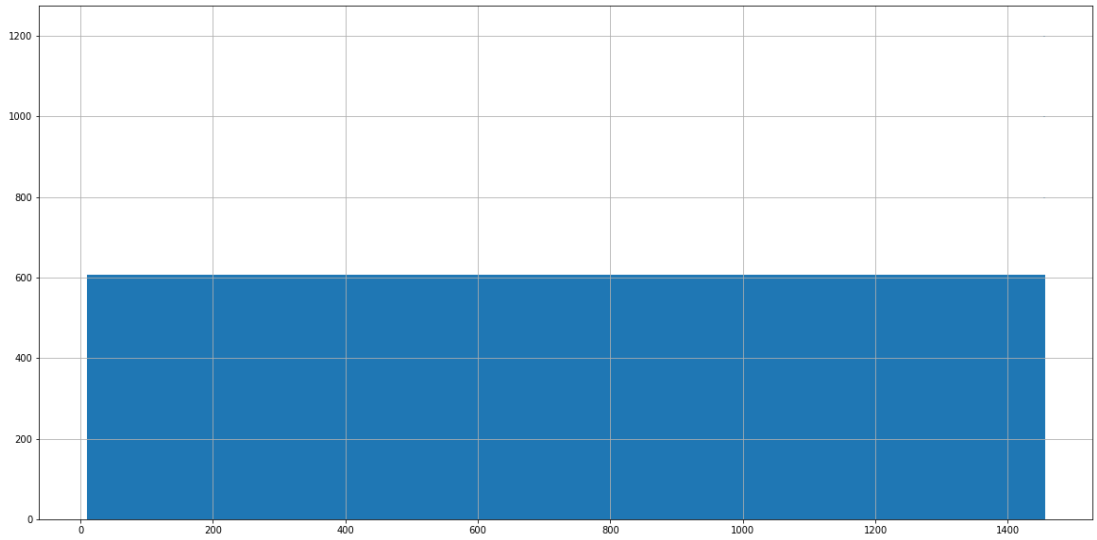


Figure 4.14: Distribution of Power Values After Oversampling.

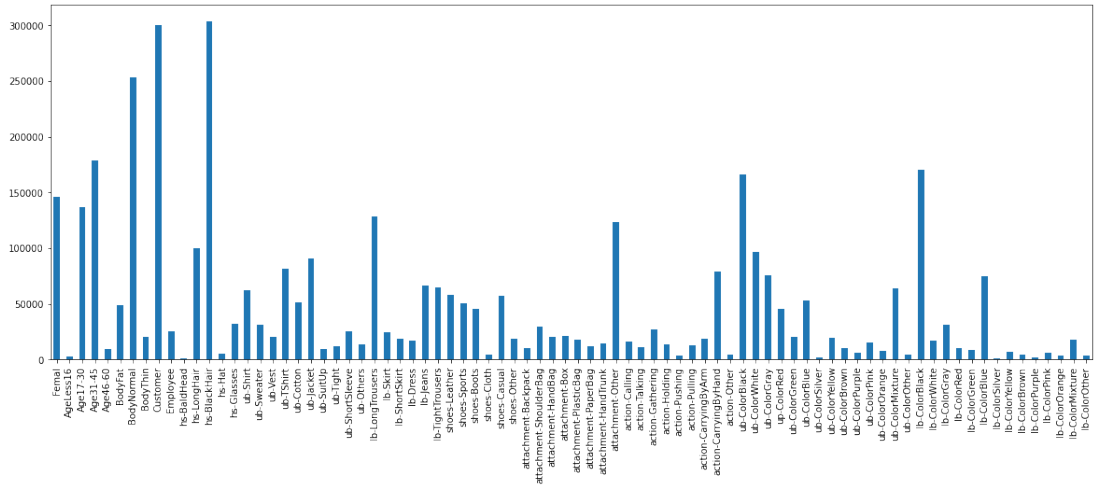


Figure 4.15: Distribution of Positive Samples for Train Data Applying Oversampling.

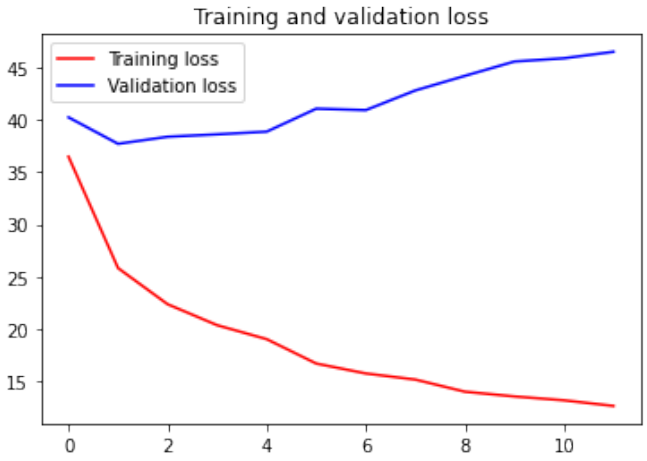


Figure 4.16: Loss of ResNet 152 V2.

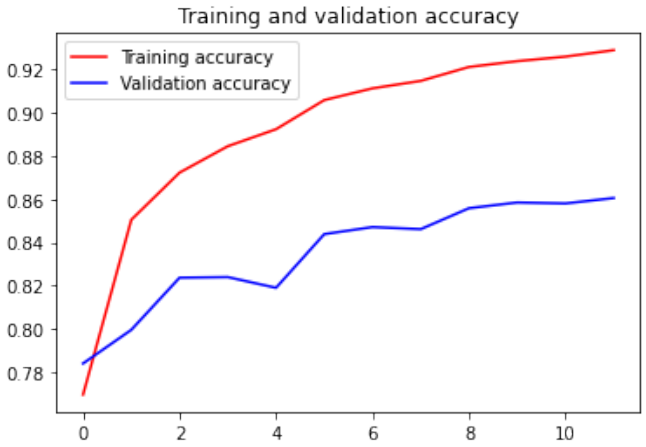


Figure 4.17: Accuracy of ResNet 152 V2.

The validation loss is increasing in this case. The weights used are from the training dataset, and the same weights are used for validation loss. But as the weights of validation data are different, it increases the validation loss. Our main observation is the training loss and training & validation accuracy.

4.5.4.2 Confusion Matrix of ResNet 152 V2 Architecture

Pedestrian attribute recognition is a multi-label classification task. As for multi-label classification, a single confusion matrix can not describe the result. So for every label, a confusion matrix is required. Keeping this in mind, we showed the confusion matrix for our proposed architecture in figure 4.18.

Figure 4.18 shows the confusion matrices used by this network to evaluate results. It can be shown that certain classes have been misclassified due to semantic similarity with the misclassified classes.

4.5.5 Imbalanced Dataset Experiment on ResNet 152 V2 with 54 Attributes

We have also experimented with 54 attributes of the RAP v2[24] dataset for a fair comparison with other existing methods. Same hyperparameters and architecture are used.

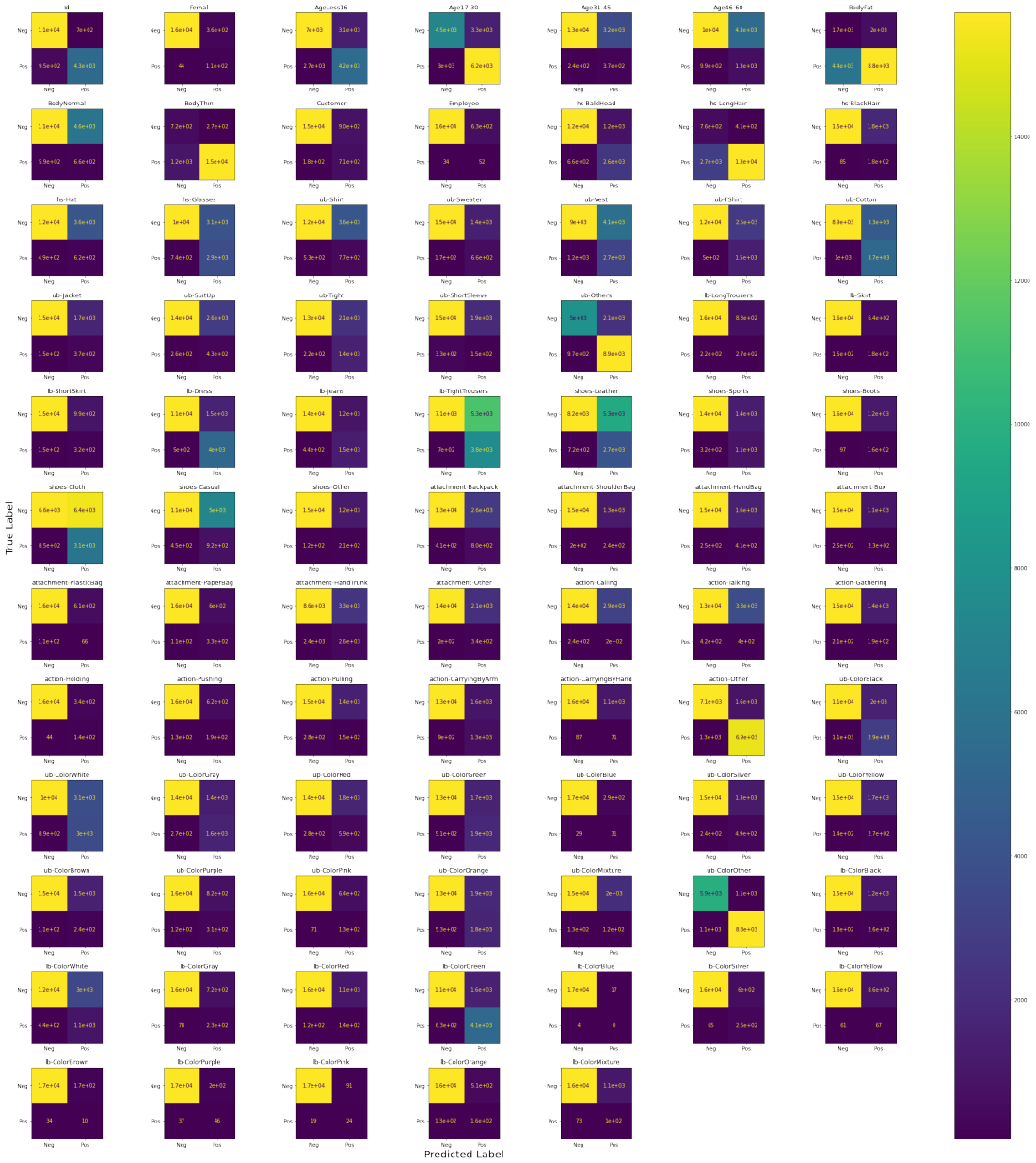


Figure 4.18: Confusion Matrix of ResNet 152 V2.

4.5.1.1 Loss and Accuracy Curve of ResNet 152 V2

The loss and accuracy curve of imbalanced dataset experiment on ResNet 152 v2 architecture with 54 attributes are shown in figure 4.19 and 4.20 respectively.

As per our hyperparameter setting, the training stopped after 35 epochs in this case.

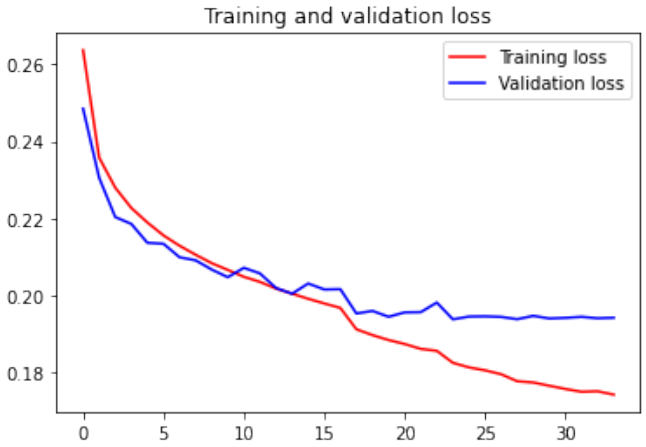


Figure 4.19: Loss of ResNet 152 V2.

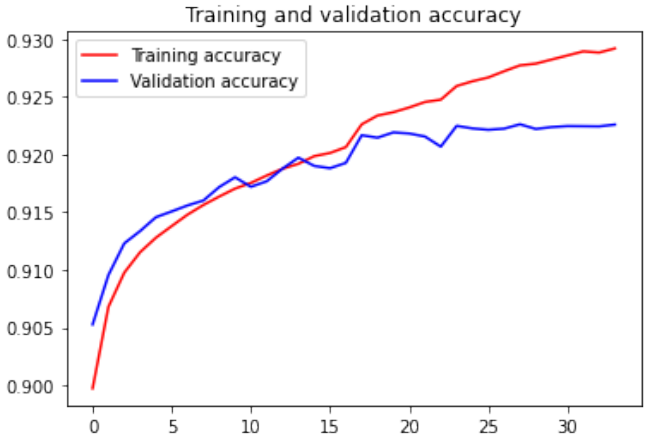


Figure 4.20: Accuracy of ResNet 152 V2.

4.5.5.2 Confusion Matrix of ResNet 152 V2 Architecture

The task of recognizing pedestrian attributes is a multi-label classification task. In the case of multi-label classification, a single confusion matrix is insufficient to explain the outcome. As a result, a confusion matrix is needed for each label. With this in mind, we built a confusion matrix for our task (see figure 4.21).

The confusion matrices used by this network to test results are shown in the figure 4.21, and it can be seen that some classes were misclassified due to semantic similarity with the misclassified classes.

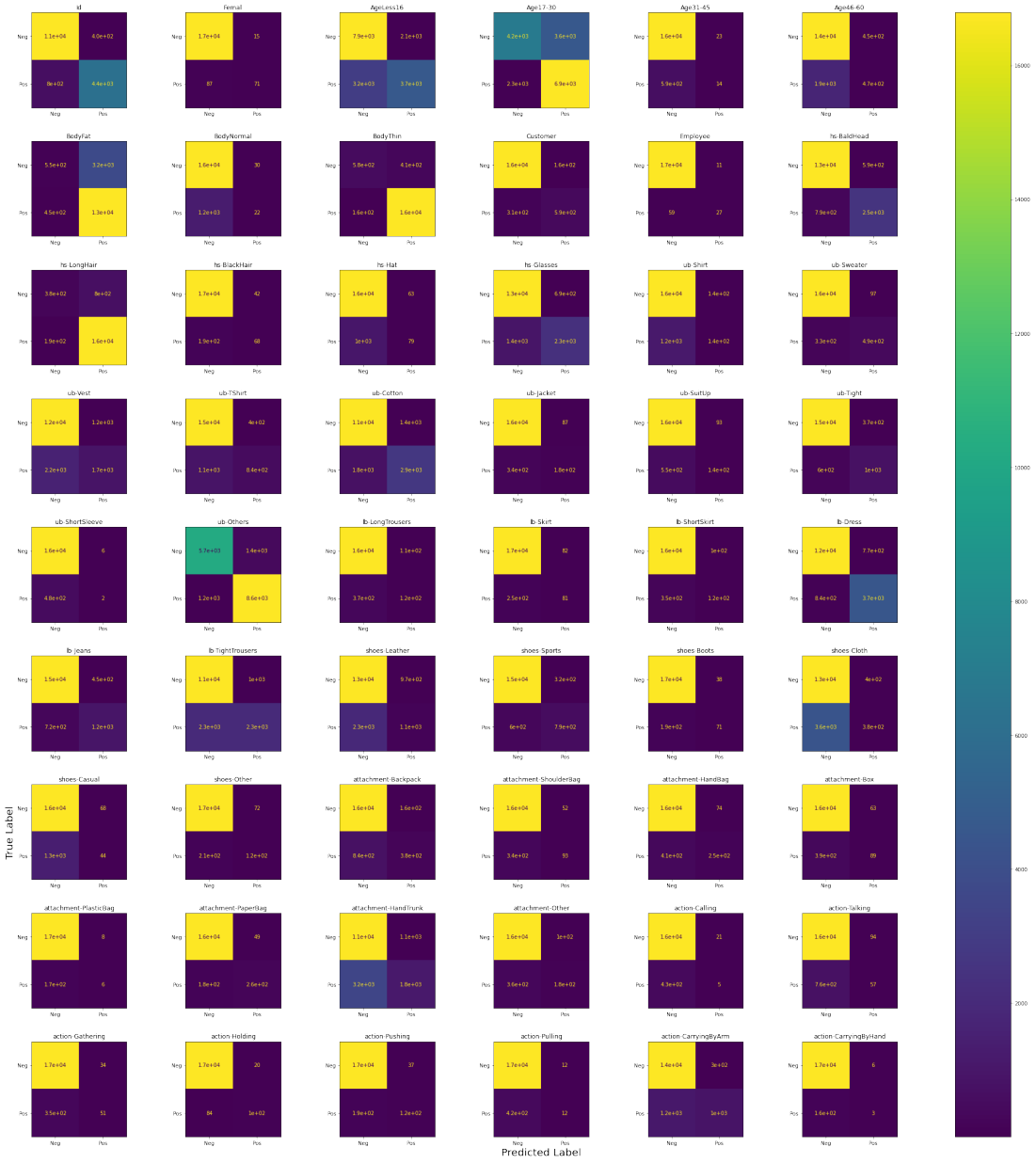


Figure 4.21: Confusion Matrix of ResNet 152 V2.

4.5.6 Balanced Dataset Experiment on ResNet 152 V2 with 54 Attributes

As stated in section 4.5.4, the experiment is conducted on 82 attributes. Similarly, in this section, we will conduct the same experiment with 54 attributes.

Figure 4.22 shows the distribution of power values before applying the oversampling technique.

After applying the oversampling technique, we get the figure 4.23. The power

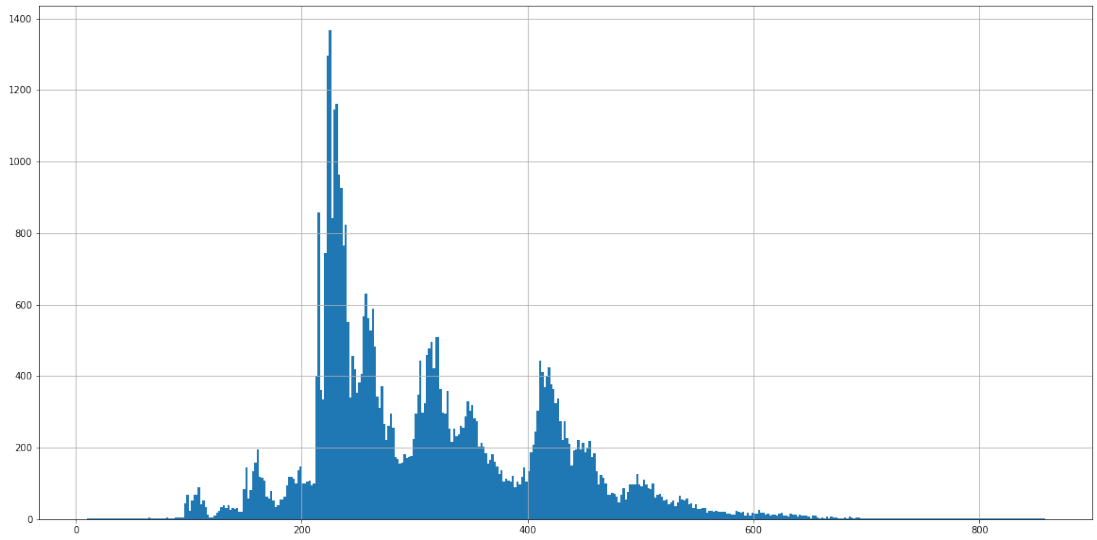


Figure 4.22: Distribution of Power Values Before Oversampling.

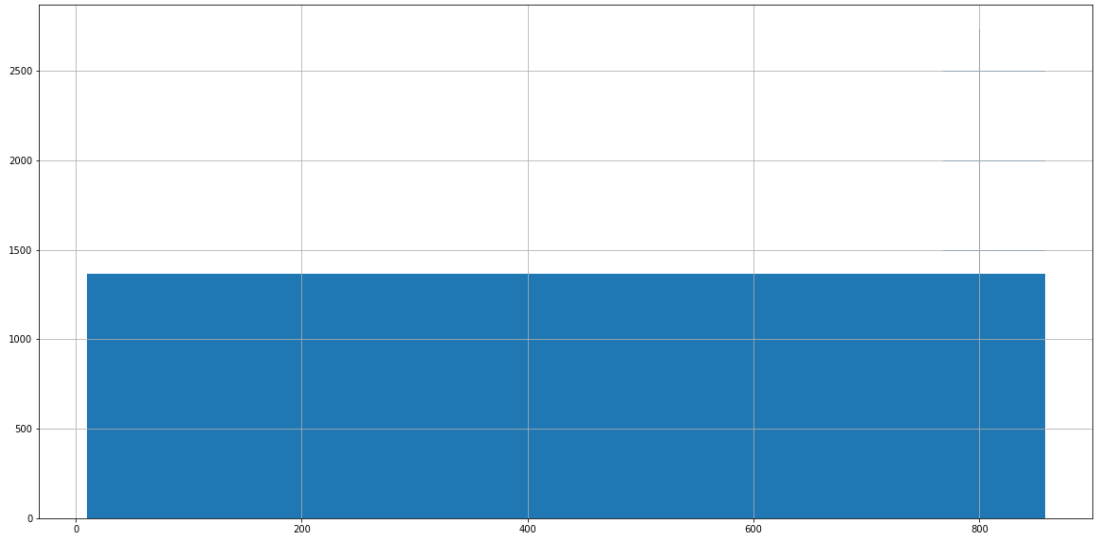


Figure 4.23: Distribution of Power Values After Oversampling.

values are now equally distributed. As this is a histogram, the x-axis represents the power values, and the y-axis represents the frequency of power values.

Finally, the positive label distribution of the training data is obtained in figure 4.24. The training data has increased significantly because of applying the oversampling technique. The total number of train images are 462046.

4.5.6.1 Loss and Accuracy Curve of ResNet 152 V2

As ResNet 152 v2 has performed better than other architectures, we conducted data balancing experiment with same hyperparameters with this architecture.

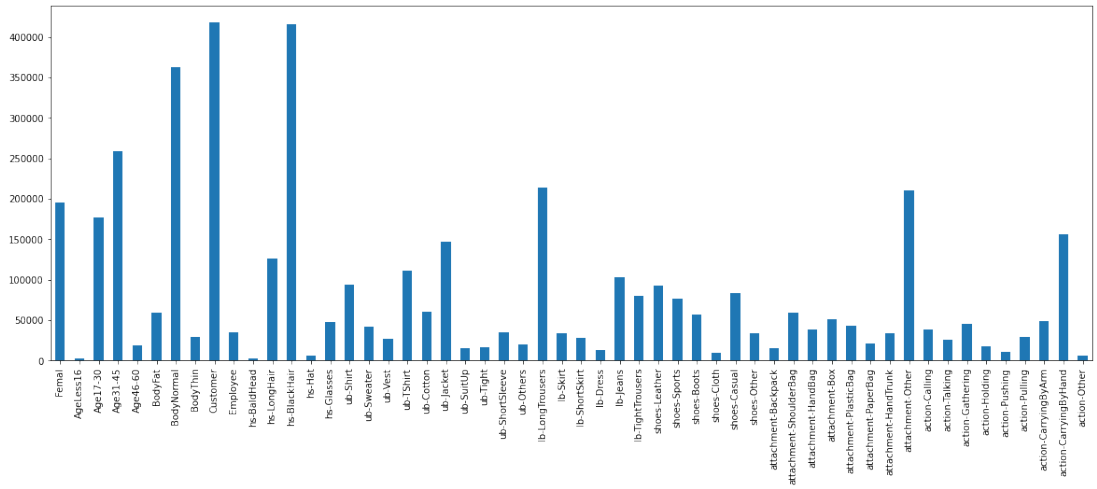


Figure 4.24: Distribution of Positive Samples for Train Data Applying Oversampling.

Applying the oversampling and weighted binary cross-entropy loss, we get the loss and accuracy curve in figure 4.25 and 4.26 respectively.

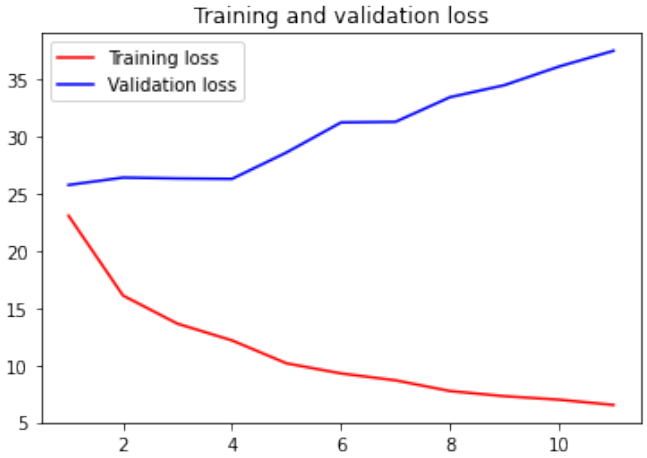


Figure 4.25: Loss of ResNet 152 V2.

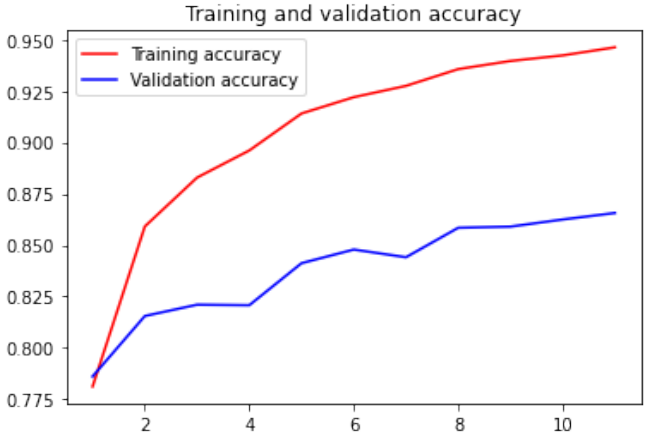


Figure 4.26: Accuracy of ResNet 152 V2.

The validation loss is increasing in this case. The weights used are from the training dataset, and the same weights are used for validation loss. But as the weights of validation data are different, it increases the validation loss. Our main observation is the training loss and training & validation accuracy.

4.5.6.2 Confusion Matrix of ResNet 152 V2 Architecture

Pedestrian attribute recognition is a multi-label classification task. As for multi-

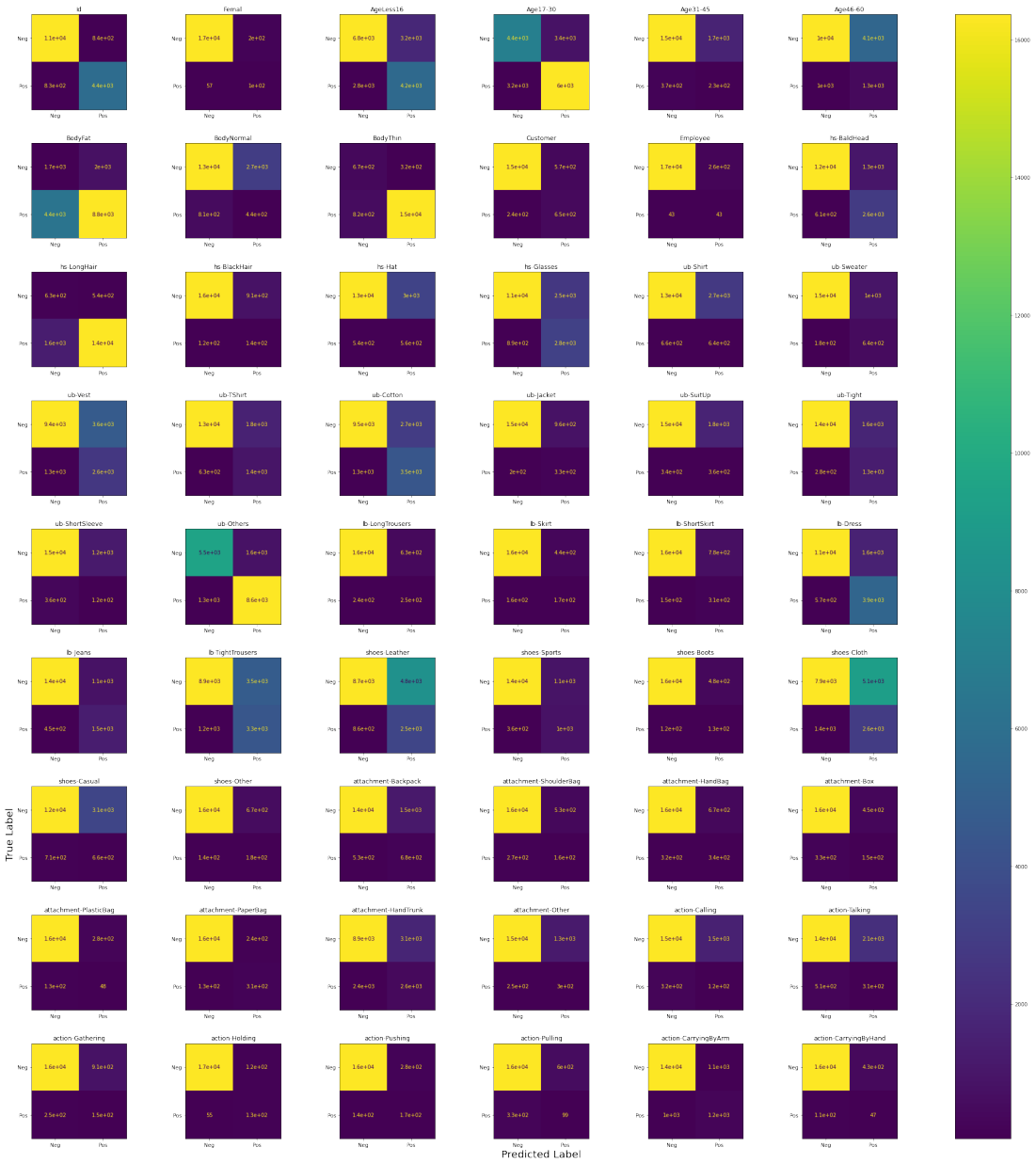


Figure 4.27: Confusion Matrix of ResNet 152 V2.

label classification, a single confusion matrix can not describe the result. So for

every label, a confusion matrix is required. Keeping this in mind, we showed the confusion matrix for our proposed architecture in figure 4.27.

Figure 4.27 shows the confusion matrices used by this network to evaluate results. It can be shown that certain classes have been misclassified due to semantic similarity with the misclassified classes.

4.6 Evaluation of Performance

After choosing the CNN architecture for the task, our next step is to compare our proposed architecture with other existing methods. But first, we are going to show a comparison among the balanced and imbalanced experiments performed in the previous sections. In this section, we will discuss the comparison with state-of-the-art networks. Also, we are going to visualise the feature map of our architecture to better understand it. And finally, the output of our overall framework.

4.6.1 Comparison Among Balanced Imbalanced Experiment Proposed Architecture

In the previous sections, we performed several experiments. Imbalanced dataset experiment with 82 and 54 attributes as well as balanced dataset experiment with 82 and 54 attributes. The side by side comparison among these experiments is shown in table 4.4.

Table 4.4: Comparison Among Balanced Imbalanced Experiment with Proposed Architecture.

Architecture Used	**Experiment Type**	**No. of Attributes**	**Trainable Parameter**	**No. of Images**	**mA**	**mP**	**mR**	**F1**
ResNet 152 V2 (last 14 trainable layers)	Balanced	82	6653010	330207	86.03	32.21	65.21	40.19
		54	6624310	462046	86.50	37.89	58.66	44.48
	Imbalanced	82	6653010	50862	**93.41**	61.34	39.15	45.18
		54	6624310	50862	92.23	63.94	39.10	45.35

4.6.2 Comparison with State-of-the-art Networks

For our proposed architecture, we selected 54 binary attributes and 2 multi-label attributes. But other existing methods didn't use the same attributes as

ours. Moreover, we selected more attributes to work with than other existing methods. We can compare with their work because we included the attributes others also used. Table 4.5 illustrates the comparison with the SOTA (state-of-the-art) methods.

Table 4.5: Comparison with state-of-the-art Methods on RAP v2 Dataset.

Method	**RAP V2 Dataset**			
	mA	**mP**	**mR**	**F1 Score**
P. Sudowe et al. [11]	68.92	70.89	80.90	75.56
D. Li et al. [10]	75.54	76.56	78.64	77.59
N. Sarafianos et al. [26]	77.87	79.03	79.79	79.04
H. Guo et al. [27]	76.74	80.42	78.78	79.24
C. Tang et al. [28]	78.21	78.25	80.43	78.93
J. Jia et al. [29]	77.34	81.99	75.62	78.21
Ours	**93.41**	61.34	39.15	45.18

In our proposed architecture, the accuracy is better than other existing methods. But for other metrics, it is not showing better results. The reason behind it this scenario is that as we selected 54 binary attributes and 2 multi-label attributes, the dataset became imbalanced than before. The distribution of the positive sample is discussed in section 4.2. For this reason, precision and recall are low in our proposed architecture as they are sensitive to the positive and negative labels. Similarly, as the f1 score depends on precision and recall, the f1 score is also low for this reason. But the overall accuracy is better than other existing methods.

4.6.3 Feature Map Visualisation

Feature maps are the extracted features from a layer. In our architecture, we have many convolution and pooling layers. To be specific, 152 layers for extracting features from an image. For simplicity, we visualised only some layers as given in the architecture of ResNet 152 V2. The feature map is visualized in figure 4.28.

We can clearly see from figure 4.28 that the first few layers are responsible for the edge detection. Image is sharpened in these layers. After the Max pool layer,

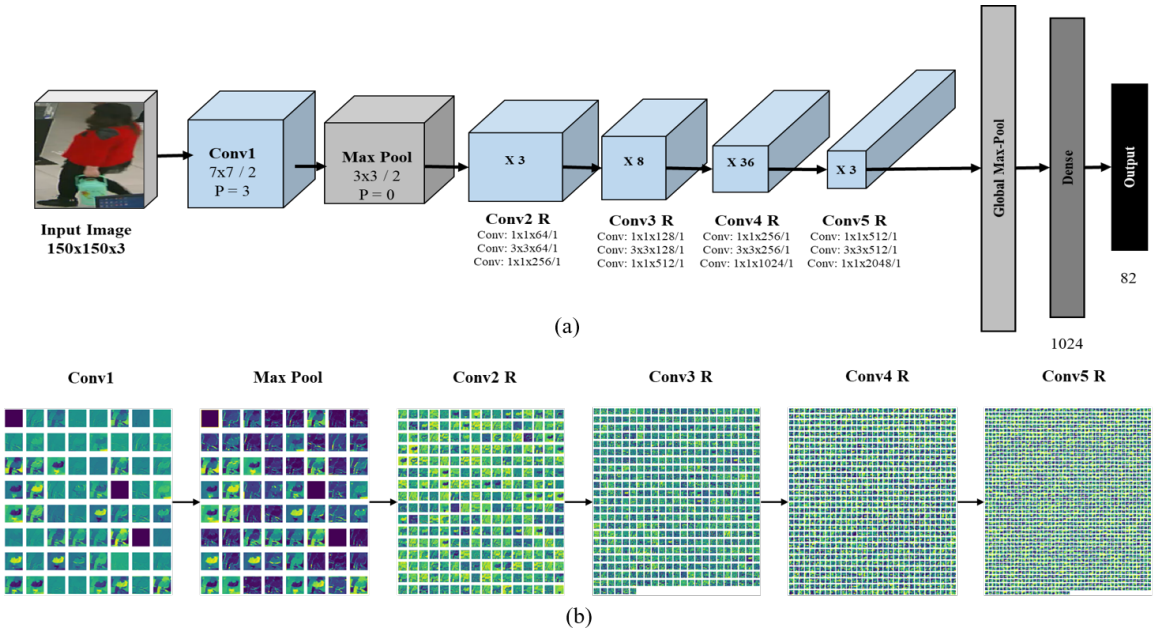


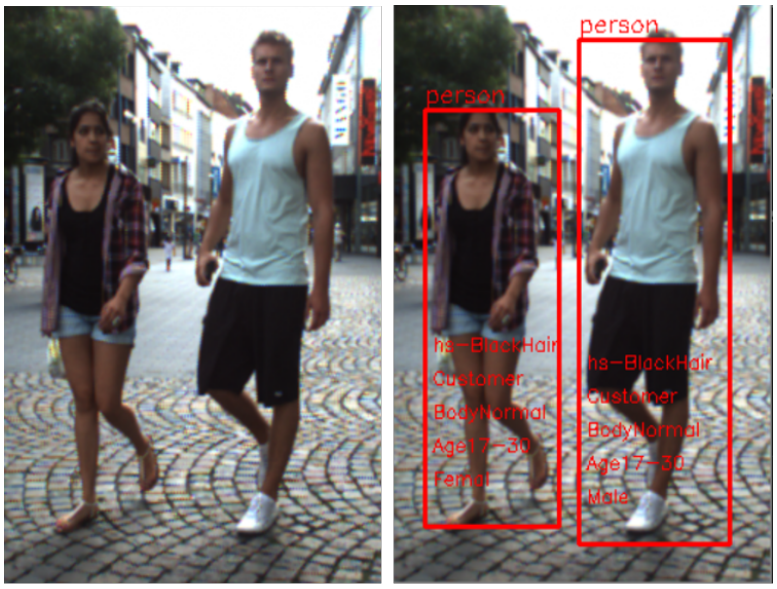
Figure 4.28: Feature Map Visualisation (a) Architecture of ResNet 152 v2 (b) Feature Map of ResNet 152 V2.

the output from the Conv2 R layer is becoming more abstract. This is the power of a convolutional neural network. It extracts features that are impossible in the human eye. The deeper the layer goes, the more abstract the image becomes. And finally, these extracted features are passed to a Fully Connected Layer for classification purpose.

4.6.4 Output of Our Proposed Framework

Our proposed framework can identify pedestrians as well as recognize pedestrian attributes from a given real-world scenario image. The demonstration of our pedestrian attribute recognition framework is shown in figure 4.29. The image is taken from another dataset in [11].

Initially, pre-processing is done as described in our framework. After that, Mask-RCNN is used to extract isolated pedestrians with bounding boxes. Each isolated pedestrians are processed before going to the next step. The processing step consists of image resizing and scaling. After completing this step, the processed isolated pedestrian image is passed to our chosen CNN architecture, ResNet 152 V2. The CNN model extracts necessary features and forward them to an FC



(a) (b)

Figure 4.29: Demonstration of our proposed framework (a) Input image (b) Output image.

layer and then to a classifier. After completing all these steps, we get the output in figure 4.29 (b).

4.7 Conclusion

The outcomes of the pedestrian attribute recognition based on a custom-designed CNN model are represented in this chapter. The performances of the different CNN architectures as well as the experimentation on the chosen architecture is also discussed. Also, a comparison among balanced & imbalanced experiment with proposed architecture is shown in this chapter. Comparison with existing methods is additionally discussed in this chapter. Feature map visualisation and the demonstration of our framework is shown here. As seen by the analysis, the proposed ResNet 152 V2 architecture outperforms existing methods by means of mA. The conclusion to this thesis work is drawn in the next chapter.

Chapter 5

Conclusion

5.1 Conclusion

The pedestrian attribute recognition task has a great impact on our society. Person re-identification, person recognition, public security etc., are the basic applications of this work. The great research area of this field is a big motivation for this work. However, this task is quite challenging because spatial feature extraction is not an easy task. Complex CNN architectures are required to capture the spatial features efficiently.

Keeping that in mind, we proposed a framework that can work on a real-world scenario. Mask-RCNN object detector extracts isolated pedestrians from an image. After that, some pre-processing is performed before passing it to our chosen CNN architecture. Extensive experiments are performed to obtain the CNN architecture that will perform efficiently.

Firstly, some pre-trained architectures are chosen based on their accuracy on the IMAGENET dataset. Then some experiments are performed on these architectures on the RAP dataset. Also, a brief analysis of the dataset is also shown in our proposed framework. After experimenting among the pre-trained architectures, a primary architecture is chosen for the next experiment. Extensive experiments are performed on the primary architecture, and finally, a CNN architecture is chosen that gives better performance than existing methods based on accuracy. Additionally, data balancing techniques are also applied to the RAP v2 dataset. Also, in our proposed architecture, we have shown the feature map of the CNN layers to get the abstract idea of how an image transforms throughout the architecture. Finally, an implementation of our proposed framework is demonstrated.

5.2 Future Work

In our proposed framework, we customised a CNN architecture for recognising pedestrian attributes, experimenting with transfer learning on different pre-trained architectures and testing on the primary architecture after it is a new approach in this field. Our approach showed better result than existing methods.

In future, we aim to work with a more balanced dataset for gaining more better performance. Also, more extensive experiments on the large dataset are also considered for our future work. Fine-tuning on the hyper-parameters can also be applied to obtain better architecture.

To enhance our proposed framework, we can consider for the person re-identification with pedestrian attributes. As person re-identification has many applications, this work will be very useful. But first of all, better accuracy in recognising pedestrian attributes is a must. In our proposed CNN architecture, we have achieved better performance which is a big contribution to the deep learning field.

References

[1] A. Khan, A. Sohail, U. Zahoora and A. S. Qureshi, 'A survey of the recent architectures of deep convolutional neural networks,' *Artificial Intelligence Review*, vol. 53, no. 8, pp. 5455–5516, 2020 (cit. on p. 2).

[2] S. Gong, M. Cristani, C. C. Loy and T. M. Hospedales, 'The re-identification challenge,' in *Person re-identification*, Springer, 2014, pp. 1–20 (cit. on p. 4).

[3] D. Gray and H. Tao, 'Viewpoint invariant pedestrian recognition with an ensemble of localized features,' in *European conference on computer vision*, Springer, 2008, pp. 262–275 (cit. on p. 7).

[4] B. J. Prosser, W.-S. Zheng, S. Gong, T. Xiang, Q. Mary *et al.*, 'Person re-identification by support vector ranking.,' in *BMVC*, vol. 2, 2010, p. 6 (cit. on p. 8).

[5] R. Layne, T. M. Hospedales, S. Gong and Q. Mary, 'Person re-identification by attributes.,' in *Bmvc*, vol. 2, 2012, p. 8 (cit. on p. 8).

[6] R. Layne, T. M. Hospedales and S. Gong, 'Attributes-based re-identification,' in *Person re-identification*, Springer, 2014, pp. 93–117 (cit. on p. 8).

[7] Y. Deng, P. Luo, C. C. Loy and X. Tang, 'Pedestrian attribute recognition at far distance,' in *Proceedings of the 22nd ACM international conference on Multimedia*, 2014, pp. 789–792 (cit. on p. 8).

[8] D. Li, Z. Zhang, X. Chen, H. Ling and K. Huang, 'A richly annotated dataset for pedestrian attribute recognition,' *arXiv preprint arXiv:1603.07054*, 2016 (cit. on pp. 8, 31).

[9] D. A. Vaquero, R. S. Feris, D. Tran, L. Brown, A. Hampapur and M. Turk, 'Attribute-based people search in surveillance environments,' in *2009 workshop on applications of computer vision (WACV)*, IEEE, 2009, pp. 1–8 (cit. on p. 8).

[10] D. Li, X. Chen and K. Huang, 'Multi-attribute learning for pedestrian attribute recognition in surveillance scenarios,' in *2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR)*, IEEE, 2015, pp. 111–115 (cit. on pp. 9, 54).

[11] P. Sudowe, H. Spitzer and B. Leibe, 'Person attribute recognition with a jointly-trained holistic cnn model,' in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2015, pp. 87–95 (cit. on pp. 9, 54, 55).

[12] J. Zhu, S. Liao, D. Yi, Z. Lei and S. Z. Li, 'Multi-label cnn based pedestrian attribute learning for soft biometrics,' in *2015 International Conference on Biometrics (ICB)*, IEEE, 2015, pp. 535–540 (cit. on p. 9).

[13] T. Matsukawa and E. Suzuki, 'Person re-identification using cnn features learned from combination of attributes,' in *2016 23rd international conference on pattern recognition (ICPR)*, IEEE, 2016, pp. 2428–2433 (cit. on p. 9).

[14] W. Fang, J. Chen and R. Hu, 'Pedestrian attributes recognition in surveillance scenarios with hierarchical multi-task cnn models,' *China Communications*, vol. 15, no. 12, pp. 208–219, 2018 (cit. on p. 9).

[15] L. Kurnianggoro and K.-H. Jo, 'Identification of pedestrian attributes using deep network,' in *IECON 2017-43rd Annual Conference of the IEEE Industrial Electronics Society*, IEEE, 2017, pp. 8503–8507 (cit. on p. 9).

[16] P. Liu, X. Liu, J. Yan and J. Shao, 'Localization guided learning for pedestrian attribute recognition,' *arXiv preprint arXiv:1808.09102*, 2018 (cit. on p. 10).

[17] K. Han, Y. Wang, H. Shu, C. Liu, C. Xu and C. Xu, 'Attribute aware pooling for pedestrian attribute recognition,' *arXiv preprint arXiv:1907.11837*, 2019 (cit. on p. 10).

[18] Z. Tan, Y. Yang, J. Wan, H. Hang, G. Guo and S. Z. Li, 'Attention-based pedestrian attribute analysis,' *IEEE transactions on image processing*, vol. 28, no. 12, pp. 6126–6140, 2019 (cit. on p. 10).

[19] Y. Li, H. Xu, M. Bian and J. Xiao, 'Attention based cnn-convlstm for pedestrian attribute recognition,' *Sensors*, vol. 20, no. 3, p. 811, 2020 (cit. on p. 10).

[20] H. Zeng, H. Ai, Z. Zhuang and L. Chen, 'Multi-task learning via co-attentive sharing for pedestrian attribute recognition,' in *2020 IEEE International Conference on Multimedia and Expo (ICME)*, IEEE, 2020, pp. 1–6 (cit. on p. 10).

[21] W. Yu, S. Kim, F. Chen and J. Choi, 'Pedestrian detection based on improved mask r-cnn algorithm,' in *International Conference on Intelligent and Fuzzy Systems*, Springer, 2020, pp. 1515–1522 (cit. on p. 13).

[22] *Github - facebookresearch/detectron: Fair's research platform for object detection research, implementing popular algorithms like mask r-cnn and retinanet.* `https://github.com/facebookresearch/Detectron`, (Accessed on 04/13/2021) (cit. on p. 13).

[23] X. Wu, S. Wen and Y.-a. Xie, 'Improvement of mask-rcnn object segmentation algorithm,' in *International Conference on Intelligent Robotics and Applications*, Springer, 2019, pp. 582–591 (cit. on p. 13).

[24] D. Li, Z. Zhang, X. Chen and K. Huang, 'A richly annotated pedestrian dataset for person retrieval in real surveillance scenarios,' *IEEE transactions on image processing*, vol. 28, no. 4, pp. 1575–1590, 2018 (cit. on pp. 15, 31, 43, 46).

[25] *Keras applications*, `https://keras.io/api/applications/`, (Accessed on 04/20/2021) (cit. on p. 22).

[26] N. Sarafianos, X. Xu and I. A. Kakadiaris, 'Deep imbalanced attribute classification using visual attention aggregation,' in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 680–697 (cit. on p. 54).

[27] H. Guo, K. Zheng, X. Fan, H. Yu and S. Wang, 'Visual attention consistency under image transforms for multi-label image classification,' in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 729–739 (cit. on p. 54).

[28] C. Tang, L. Sheng, Z. Zhang and X. Hu, 'Improving pedestrian attribute recognition with weakly-supervised multi-scale attribute-specific localization,' in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 4997–5006 (cit. on p. 54).

[29] J. Jia, H. Huang, W. Yang, X. Chen and K. Huang, 'Rethinking of pedestrian attribute recognition: Realistic datasets with efficient method,' *arXiv preprint arXiv:2005.11909*, 2020 (cit. on p. 54).