FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Bias detection and mitigation for Biometrics Explainable Artificial Intelligence methods applied to Face Recognition

Mafalda Luisa Sampaio Oliveira



Master in Bioengineering - Biomedical Engineering

Supervisor: Ana Filipa Sequeira, PhD

Co-supervisor: Pedro David Carneiro Neto, MSc

Co-supervisor: Tiago Filipe Sousa Gonçalves, MSc

October 17, 2024

© Mafalda Luisa Sampaio Oliveira, 2024

Bias detection and mitigation for Biometrics Explainable Artificial Intelligence methods applied to Face Recognition

Mafalda Luisa Sampaio Oliveira

Master in Bioengineering - Biomedical Engineering

October 17, 2024

Resumo

Com o surgimento de técnicas de *deep learning*, houve uma grande melhoria da performance de diferentes tarefas. Contudo, este aumento na performance está associado à redução da transparência dos modelos devido ao elevado número de parâmetros dos modelos treinados. Estes modelos são considerados modelos "caixa-negra" porque reduzem a confiança do utilizador e levam a situações indesejadas em aplicações que trabalham com dados sensíveis, como aplicações biométricas.

Os sistemas de reconhecimento facial determinam a identidade de uma pessoa através da sua face. Estes sistemas têm diversas aplicações, como o controlo de fronteiras e segurança pública. Tal como noutras áreas, o uso de *deep learning* nestes sistemas levou a um grande aumento na sua performance associado a uma menor transparência destes modelos. Alguns estudos demonstraram que estes sistemas têm diferentes desempenhos para diferentes grupos demográficos.

Inteligência artificial explicável é uma área que tem como objetivo criar modelos mais transparentes que mantenham a alta performance. O reconhecimento facial explicável é uma área que pretende tornar explicável a razão de duas imagens faciais serem identificadas como a mesma pessoa ou não. Assim, a explicabilidade em reconhecimento facial permite aumentar a confiança do utilizador e a deteção e mitigação de viés presente nestes sistemas.

O presente trabalho foi desenvolvido com base num projeto cujo objetivo final seria desenvolver um método de reconhecimento facial explicável. Particularmente, o trabalho teve como foco o desenvolvimento de um classificador de atributos faciais e o uso do método *Shapley Additive Explanations* para localizar as features que tinham um maior impacto nas previsões do modelo. Deste modo, quatro classificadores diferentes foram implementados para prever vários atributos faciais, tendo como *input templates* faciais. A diferença entre estes modelos corresponde à dimensão e origem do input (128D ou 512D, extraídos com *FaceNet* ou *iResNet100*) ou com ao número de atributos a serem classificados (2 ou 9 attributos faciais). Uma abordagem *multi-task* foi usada tendo em consideração as correlações entre atributos. Estes classificadores demonstraram uma boa performance na previsão dos atributos faciais. Para a previsão do género e etnia, o modelo que foi treinado com os *embeddings* de 512D demonstrou melhores resultados, mas para a previsão de 9 atributos faciais, o treino com os mesmos levou a pior performance. A performance da etnia e do género não mudou significativamente nos modelos treinados para prever mais atributos em comparação com os modelos que prevêm apenas género e etnia. Para alguns atributos, a performance das diferentes classes não foi igual.

Finalmente, os valores das *Shapley Additive Explanations* foram implementados para avaliar o impacto das diferentes *embedding features* na previsão final dos classificadores. Os modelos treinados com o mesmo tamanho de *embeddings* demonstraram uma ordem de importância de *features* semelhante, apesar da diferença no número de atributos classificados. Adicionalmente, diferentes atributos podem ter *features* importantes comuns, que pode estar relacionado com o uso de uma abordagem *multi-task*. As *features* mais importantes para os diferentes atributos estão espalhadas em termos de localização no *embedding*. Por outro lado, *features* que estão localizadas proximamente no *embedding* podem ter impactos diferentes na decisão do modelo.

Abstract

The emergence of deep learning led to a great improvement in the performance of different tasks. Nevertheless, this increase was associated with the lack of transparency in the models due to the high number of parameters of trained models. These "black-box" models reduce the users' trust and can lead to undesired situations in applications that deal with sensitive data, such as biometrics.

Facial recognition systems use the face to identify a person's identity. These systems have several applications such as law enforcement, border control and smartphone unlock. Similar to other fields, the use of deep learning boosted the performance of these systems but increased their opaqueness. Some studies have shown that these models are biased towards certain demographic groups.

Explainable artificial intelligence is an emerging field aiming to create more transparent models while maintaining high performance. The introduction of explainability in facial recognition is needed since it increases the users' trust and provides tools to detect and mitigate bias present in these models. Thus, the field of explainable face recognition aims to explain the reason behind the face matching decisions in a face recognition system.

The work presented was part of a project whose final goal would be to create a new explainable face recognition method. Particularly, the goal was to implement a face attribute classifier and use the *Shapley Additive Explanations* to locate the features that had a higher impact on the model predictions. Hence, four different classifiers were successfully implemented that predicted several attributes from face templates. The difference between these classifiers is either the embedding size and origin (128 or 512 dimensions, extracted with *FaceNet* or *iResNet100*) or the number of features being predicted (2 or 9 face attributes). A multi-task framework was used to leverage the correlations between the different attributes. These classifiers showed good performance for the different attributes. For the prediction of gender and ethnicity, the model trained with a higher dimension embedding showed better results. Nevertheless, when predicting 9 face attributes, the performance was poorer for the model trained with a higher dimension embedding. Additionally, the performance for ethnicity and gender did not significantly change when adding more face attributes. Moreover, for some attributes, the performance was different for the different classes (e.g., Asian class had a higher performance than other classes for ethnicity prediction).

Furthermore, *Shapley Additive Explanations* values were implemented to assess the impact of the different features of the embedding on the final model prediction. With summary plots, it was possible to verify for models trained with the same embedding size, but that predicted a different number of face attributes, the performance for ethnicity and gender was similar. Additionally, different attributes have common important features, that can be related to the use of the multi-task framework. The most important features for the prediction of each attribute are spread across the embedding and nearly located features in the face embedding can have different impacts on the model decision.

Agradecimentos

Primeiramente, gostaria de agradecer aos meus orientadores Ana Filipa Sequeira, Pedro Neto e Tiago Gonçalves pela oportunidade e apoio durante a realização da Dissertação.

Ao meu pai, por sempre me ter ensinado a aproveitar as 24 horas do dia e a acreditar que sou capaz de fazer tudo aquilo que defina como objetivo.

À minha mãe, por fomentar o gosto pelo conhecimento e pelo apoio durante o percurso académico.

Ao meu irmão, por todas as explicações com desenhos no *paint* e por me fazer companhia nas noitadas de trabalho.

À minha família: Paulo, Marta, Tomás, Rita, Jorge, Aniella, Mariana, Graça, Xavier, Avó Ivone, Avó Judite, Avô Salvador, Avô Artur, por toda a felicidade, apoio e carinho que me trazem.

Aos meus colegas e amigos por tornarem estes 5 anos mais fáceis e por criarem memórias para a vida.

Ao Martim pelo apoio e paciência.

Mafalda Oliveira

"Nothing in life is to be feared, it is only to be understood"

Marie Curie

Contents

1 Introduction **1**
- 1.1 Background and Motivation . . . 1
- 1.2 Objectives and Contributions . . . 2
  - 1.2.1 United Nations Sustainable Development Goals . . . 3
- 1.3 Dissertation Outline . . . 4

2 Literature Review **5**
- 2.1 Deep Learning . . . 5
  - 2.1.1 Introduction . . . 5
  - 2.1.2 History . . . 6
  - 2.1.3 Supervised Learning . . . 7
  - 2.1.4 Metrics for Classification . . . 14
  - 2.1.5 Challenges and Future Directions . . . 15
- 2.2 Face Recognition . . . 16
  - 2.2.1 Pipeline . . . 16
  - 2.2.2 Databases . . . 18
  - 2.2.3 Loss Functions . . . 18
  - 2.2.4 Network architecture . . . 21
  - 2.2.5 Challenges and Future Directions . . . 23
- 2.3 Soft-biometrics/Face Attributes Classification . . . 25
  - 2.3.1 Databases . . . 26
  - 2.3.2 Multi-task Learning . . . 26
  - 2.3.3 Multi-Attribute Estimation . . . 26
- 2.4 Explainable Artificial Intelligence . . . 30
  - 2.4.1 Introduction . . . 30
  - 2.4.2 Terminology . . . 32
  - 2.4.3 Taxonomy . . . 32
  - 2.4.4 Evaluation of Interpretability . . . 35
  - 2.4.5 Shapley Additive Explanation . . . 35
  - 2.4.6 Saliency Maps . . . 36
  - 2.4.7 Explainability in Face Recognition . . . 37

3 Face Attributes Classification **39**
- 3.1 Methodology . . . 39
  - 3.1.1 Dataset . . . 39
  - 3.1.2 Pre-processing . . . 42
- 3.2 Classifier Architecture Experiments . . . 43
  - 3.2.1 Ablation Studies and Hyperparameter Tuning: Results and Discussion . . . 44

3.2.2 Conclusions 46
3.3 Multi-task Classifiers Implementation 46
3.3.1 MTL1 47
3.3.2 MTL2 50

4 Explaining Soft-Biometrics Predictions using SHAP Values **55**
4.1 MTL1A 55
4.2 MTL1B 57
4.3 MTL2A 58
4.4 MTL2B 60
4.5 Conclusions 62

5 Conclusions and Future Work **69**
5.1 Conclusions 69
5.2 Future Work 70

References **71**

List of Figures

1.1 Pipeline for the developed work. . . . 2
1.2 17 SDGs, from [1]. . . . 3

2.1 The hierarchy of AI, ML and DL, adapted from [2]. . . . 5
2.2 DL and ML performance with the increase in the data, from [2]. . . . 6
2.3 Deep and Shallow Neural Network architectures, from [3]. . . . 9
2.4 CNN Architecture, from [3]. . . . 9
2.5 Recurrent Networks, from [4]. . . . 11
2.6 The effect of using dropout in CNNs, from [5]. . . . 11
2.7 Error rates for the different architectures proposed for ImageNet challenge, from [6]. 12
2.8 AlexNet Architecture, from [6]. . . . 12
2.9 VGGNet Architecture, from from [6]. . . . 12
2.10 Inception Block Architecture Architecture, from [6]. . . . 13
2.11 SE block architecture, from [6]. . . . 13
2.12 Example of a confusion matrix for a multi-classification problem, from [7]. . . . 15
2.13 Deep FR Pipeline, from [8]. . . . 17
2.14 Evolution of FR losses, from [8]. . . . 19
2.15 Evolution of Object Classification DL Architectures and FR Networks, from [8]. The same colour represents the same architecture. . . . 21
2.16 FaceNet architecture, from [9]. . . . 23
2.17 MTL and STL, from [10]. . . . 27
2.18 MAADFace annotation pipeline, from [11]. . . . 28
2.19 Annotation Example from MAAD-face database, from [11]. . . . 30
2.20 A- Ladder of causality, B- xAI ladder, from [12]. . . . 34
2.21 SHAP summary plot example, from [13]. . . . 36

3.1 Pipeline for the developed work. . . . 40
3.2 Correlation Matrix for the 47 attributes extracted with MAAD in BUPT-Balancedface. 41
3.3 Ethnicity distribution in BUPT-Balancedface dataset. . . . 41
3.4 Distribution for the selected attributes of the annotations. . . . 42
3.5 Correlation Matrix for the attributes used for classification. . . . 43
3.6 Ablation studies for the selection of the architecture for the classifier. . . . 44
3.7 Histogram of the accuracies per identity for MTL1A. . . . 47
3.8 Histogram of the accuracies per identity for MTL1B. . . . 48
3.9 Confusion matrix for gender prediction with MTL1A. . . . 49
3.10 Confusion matrix for gender prediction with MTL1B. . . . 49
3.11 Confusion matrix for ethnicity prediction with MTLA. . . . 49
3.12 Confusion matrix for ethnicity prediction with MTLB. . . . 49

3.13 Confusion matrix for multiple attribute prediction with MTL2A. 52
3.14 Confusion matrix for multiple attribute prediction with MTL2B. 53

4.1 SHAP Summary plot for gender prediction of MTL1A. 56
4.2 SHAP Summary plot for ethnicity prediction of MTL1A. 56
4.3 SHAP Summary plot for gender prediction of MTL1B. 57
4.4 SHAP Summary plot for ethnicity prediction of MTL1B. 58
4.5 SHAP Summary plot for gender prediction of MTL2A. 59
4.6 SHAP Summary plot for ethnicity prediction of MTL2A. 59
4.7 SHAP Summary plot for face shape prediction with MTL2A. 61
4.8 SHAP Summary plot for High Cheekbones prediction with MTL2A. 62
4.9 SHAP Summary plot for nose attributes prediction with MTL2A. 63
4.10 SHAP Summary plot for lip shape prediction with MTL2A. 63
4.11 SHAP Summary plot for gender prediction of MTL2B. 64
4.12 SHAP Summary plot for ethnicity prediction of MTL2B. 64
4.13 SHAP Summary plot for face shape prediction for MTL2B. 65
4.14 SHAP Summary plot for high cheekbone prediction for MTL2B. 66
4.15 SHAP Summary plot for lip shape prediction for MTL2B. 66
4.16 SHAP Summary plot for nose shape prediction for MTL2B. 67

List of Tables

2.1 Decision Boundary for binary case for class 1, with s being the normalised features. 20
2.2 Performance of FR methods in LFW dataset, adapted from [8]. Acc and Std correspond to accuracy and standard deviation, respectively. 24
2.3 Summary of Methods for Multi-Attribute Estimation. Avg is an abbreviation for average. 29

3.1 Architecture A Experiments. 45
3.2 Batch Size Experiments. 45
3.3 Architecture B Experiments 45
3.4 Architecture C Experiments. 46
3.5 Architecture D Experiments. 46
3.6 Comparison between MTL1A and MTLB results. 47
3.7 MTL2A performance per attribute. 51
3.8 MTL2B performance per attribute. 51

Abreviaturas e Símbolos

AI	Artificial Intelligence
ANN	Artificial Neural Network
BatchNorm	Batch Normalisation
CAM	Class Activation Mapping
CNN	Convolutional Neural Network
cEBP	Contrastive Excitation Backprop
Deep hidden IDentity features	DeepID
Deep IDentification-verification features	DeepID2
DL	Deep Learning
EBP	Excitation Backprop
FC	Fully-Connected
FP	False Positive
FR	Face Recognition
GPUs	Graphic Processing Units
Gradient Activation Mapping	GradCAM
ImageNet Large Scale Visual Recognition Challenge	ILSVRC
LR	Learning Rate
MAAD	Massively Annotated Attribute Dataset
MAC	Massive Attribute Classifier
ML	Machine Learning
MP	Max-Pooling
MTL	Multi-Task Learning
NLP	Natural Language Processing
ReLU	Rectified Linear Unit
Restricted Boltzmann machine	MT-RBM
RISE	Randomized Input for Explanation
RNN	Recursive Neural Network
SHAP	Shapley Additive Explanation
STL	Single Task Learning
TN	True Negative
TP	True Positive
tcEBP	Truncated Contrastive Excitation Backprop
xAI	Explainable Artificial Intelligence
xFR	Explainable Face Recognition
xSSAB	Similarity Score Argument Backpropagation

Chapter 1

Introduction

1.1 Background and Motivation

Face recognition (FR) systems predict a person's identity through a face image or video [14]. These systems have several applications such as border control, smartphone unlock, forensics, law enforcement and home security [14, 15, 16]. The emergence of deep learning (DL) increased FR performance, along with the higher availability of data, the evolution of training losses and model architectures [17, 18]. Nevertheless, these DL systems are less interpretable due to the high number of parameters of the trained models and their complexity [18].

The "black-box" nature of DL models makes the data embedded in the face templates, which are compact vectors obtained with FR algorithms, unknown. Some literature has shown that information on demographics (e.g., age, gender and ethnicity), pose, social traits and the image properties can be present in these embeddings. Studying which information is present in these templates is important for protecting user's privacy and detecting bias in the model [19].

The goal of eXplainable Artificial Intelligence (xAI) is to create more transparent models while maintaining a high performance [20, 21, 22]. Explainable face recognition (xFR) has the objective of explaining the reason behind the match or not between two images in FR systems. The introduction of explainability in these systems can help to improve their performance and increase the users' acceptance and trust [17, 18].

Saliency maps are one common research direction for xAI in computer vision, where the input feature importance for the model output is highlighted. Nevertheless, when applying this type of method for FR, there is a need for adaptation due to its differences from classification tasks [17, 18, 23]. Besides saliency maps, explainability should focus on explaining the reason behind the face matching rather than just the features that had most influence [17]. Other works in FR have focused on changing the training of FR models by training rule priors, regularisation and constraints. However, there is less investigation of explainability for biometrics when compared to other areas of computer vision. Hence, the work developed for other areas should be studied and adapted for biometrics [12].

Therefore, there is a need to increase the research in xFR methods, introducing new explainability approaches to better understand the reasons behind the functioning of these systems.

With the knowledge that different face attributes are encoded in the face templates, a new xFR method could be introduced that explains the matching decision for two images through the differences in the information encoded in their embeddings. Hence, new approaches could be related to the differences in the latent space rather than the input space.

1.2 Objectives and Contributions

The work of this dissertation was carried out as part of the "Explainable Artificial Intelligence for Biometrics (XAIBIO)" seed project, led by the Visual Computing and Machine Intelligence (VCMI) research group of the Centre for Telecommunications and Multimedia of the Institute for Systems and Computer Engineering, Technology and Science (INESC TEC). The work was financed by Fundação para a Ciência e a Tecnologia (FCT), as part of the project UIDB/50014/2020[1]. One of the goals of this project was to develop an approach for xAI applied to FR. In particular, the work developed for this dissertation had as an objective the implementation of a face attribute classifier and the computation of Shapley Additive Explanations (SHAP) values to locate the features in the embeddings that had a higher importance for the classifier's prediction. Moreover, the end goal of the project was to build an explanation method based on these soft-biometrics presence and location for FR.

The different steps necessary for the developed work are defined in Figure 1.1. The work focused mainly on the implementation of the different face attributes classifiers and the explainability module. The extraction of the labels, through the automatic annotation pipeline, and the embeddings benefited from the contributions of other members of the VCMI group.

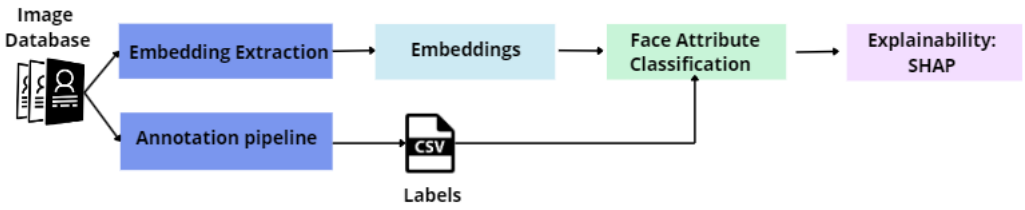


Figure 1.1: Pipeline for the developed work.

The main contributions of this dissertation include the implementation of two different multi-task classifier architectures for the prediction of gender and ethnicity and the prediction of nine different face attributes. With these classifiers, it was possible to successfully predict attributes related to gender, ethnicity, face, nose and mouth shape from face templates. Additionally, the different features' contribution to the final predictions and their impact on the model's output were analysed using SHAP values. This work is the first step for the project's final goal, i.e., a new explanation method for FR based on soft-biometrics.

[1]DOI 10.54499/UIDB/50014/2020 | https://doi.org/10.54499/uidb/50014/2020

1.2.1 United Nations Sustainable Development Goals

The United Nations Sustainable Development Goals (SDGs) [1] are 17 objectives adopted by the United Nations, as part of the "2030 Agenda for Sustainable Development", a shared program whose goal is that humanity and the planet have peace and prosperity in the present and for the future [1]. These goals are displayed in Figure 1.2. The work of this thesis is framed within different SDGs[2]:



Figure 1.2: 17 SDGs, from [1].

- **Goal 5: Gender Equality** - the study of xAI in FR can contribute to the detection and mitigation of the bias present in FR systems towards female gender.
- **Goal 9: Industry, Innovation, and Infrastructure** - xAI can create innovation by making automated systems more transparent.
- **Goal 11: Sustainable Cities and Communities** - The introduction of explainability in FR can make these methods more robust to be implemented in smart cities, for law enforcement, surveillance, border control and house security.
- **Goal 16: Peace, Justice, and Strong Institutions** - xAI can help FR systems be safer to be used for law enforcement and forensic applications, helping to achieve justice.

[2]The content of this publication has not been approved by the United Nations and does not reflect the views of the United Nations or its officials or Member States.

1.3 Dissertation Outline

This dissertation is composed of five chapters. The present chapter provides the background and the motivation that led to the pursuit of this research. Additionally, it defines the main initial goals and the contributions brought by the developed work.

Chapter 2 gives insights into the literature and the key concepts of DL, FR and xAI.

Chapter 3 reports the methodology for implementing multi-attribute face classifiers, their performance and an analysis of the main results.

Chapter 4 displays the results for the SHAP values applied to the implemented classifiers and the evaluation of these results.

Chapter 5 concludes this dissertation and provides future work directions.

Chapter 2

Literature Review

This chapter provides an extensive overview of DL, FR, face attribute classification and xAI. Section 2.1 defines the key topics of DL, the common DL architectures used for classification tasks and the main challenges that exist within this field. Section 2.2 details the main concepts of FR, FR databases, the functioning of FR systems, the evolution of the losses and architectures and the main challenges that FR still faces. Section 2.3 details some concepts of face attribute classification and the use of multi-task frameworks for this process. Finally, Section 2.4 approaches xAI terminology, taxonomy, evaluation and sub-field of xFR.

2.1 Deep Learning

2.1.1 Introduction

Artificial intelligence (AI), machine learning (ML), and DL are current trendy topics, regularly used as synonyms to define technologies that appear to have intelligent behaviour. However, these concepts are distinct [2].

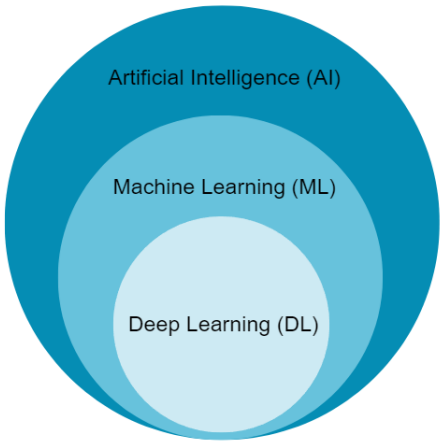


Figure 2.1: The hierarchy of AI, ML and DL, adapted from [2].

AI defines technologies that introduce human behaviour and intellectual capacity into machines or systems. ML is part of the large area of AI, it extracts knowledge from data to create analytical models in an automated way [2]. Furthermore, DL is a subdivision of ML that defines neural networks which automatically extract features from the data with different abstraction levels through several layers, hence the term "deep" [2, 4, 24]. More specifically, low-level features are extracted from the first layers, whereas the high-level features are obtained from the final layers [6]. Figure 2.1 shows the hierarchy of these terms.

DL has become the preferred approach of ML due to its impressive results on a variety of demanding challenges [6]. Additionally, contrary to traditional ML models that require domain knowledge and feature engineering to extract a feature representation from raw data, DL models do this in an automated and faster way, without human expertise and effort [2, 4, 6, 25].

The performance of DL algorithms increases exponentially with the data quantity, whereas the results from traditional ML tend to stabilise (see Figure 2.2) [2, 3]. Compared to traditional ML models, DL models usually show faster testing times, even though their large number of parameters means they take longer to train. To handle the computations related to DL training, greater computational resources are used, such as Graphics Processing Units (GPUs) [2].

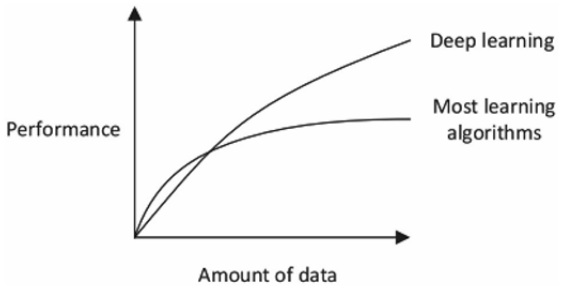


Figure 2.2: DL and ML performance with the increase in the data, from [2].

2.1.2 History

DL history began in 1943 with the proposal of McCulloch-Pitts model, which was based on the functioning of biological neural networks. It used a threshold logic to simulate the thought process. Later, in 1949, the Hebbian theory defined that a neural pathway is reinforced every time it is used. In 1958, Frank Rosenblatt introduced the "perceptron" integrated into a machine, which is different from the modern perceptron concept. It was a single-layered perceptron that performed binary classification in single-valued inputs, applying an input-weighted sum and then a threshold to determine one of the two possible values [3, 25]. After that, backpropagation and error-based learning was introduced and it paved the way for modern neural networks [25]. In the eighties, the network "neocognitron" that inspired the convolutional neural networks was proposed, along with Recurrent Neural Networks (RNNs). In the 1990s, the LeNet, a basic convolutional neural network, was introduced. Even though neural networks had some applications, they could not be applied to large datasets due to hardware constraints and they were overlooked [3, 25]. The

general belief was that it was impracticable to learn an automatic feature extractor with little domain knowledge [25].

Despite these early advancements, the term DL only became popular after 2006, with the work introduced by Hilton et al. [26]. The authors proposed a pre-training technique for restricted Boltzmann machines. This technique contributed to a substantial increase in the depth of the models and a new generation of neural networks emerged [2, 25]. Additionally, the introduction of GPUs led to an increase in the training times of these models. Further research showed that the pre-training technique is useful for small datasets. During this period of sucess for DL, Convolutional Neural Networks (CNNs) achieved great results and were adopted in computer vision research [4].

When properly trained, CNNs have accomplished great performance in regression and classification problems [25]. DL is now one of the most important tools in the areas of ML, AI, data science and analytics [2, 25]. Furthermore, the industry is focusing on creating automated intelligent systems and DL is becoming the core method for this evolution [2]. Technology and economy-based companies are competing to improve DL methods [6].

2.1.3 Supervised Learning

ML algorithms can be categorised into: Supervised, Unsupervised, Semi-supervised and Reinforcement Learning. Supervised Learning is an approach that tries to solve a specific task using labels for each example in the dataset to guide the learning. By contrast, Unsupervised Learning algorithms learn useful information from unlabelled datasets by capturing the structure and patterns in the data. Semi-supervised learning merges the two previous approaches. Finally, in Reinforcement Learning techniques, a software agent learns how to efficiently make decisions stimulated by a specific environment [2, 6, 24, 27].

Supervised learning is the most popular approach to ML, being deep or not. This type of learning requires the use of large labelled datasets. In the case of a classification task, for a certain input, the model generates a vector of scores with the size of the number of possible outputs, and the one that corresponds to the true label should have the highest score. This is only possible after the training of the model [4].

A loss function that measures the dissimilarity between the labels and the predictions is used for supervising the training of the model [4]. Cross-entropy loss is commonly applied to measure the performance in classification tasks. The formula for this loss is represented in Equation 2.1. N corresponds to the number of output neurons, p is the output probability and y is the true label [6].

$$H(p,y) = -\sum_{i} y_i \log(p_i) \quad , \quad i \in [1,N] \tag{2.1}$$

To adjust the model to produce the results close to the labels, the model parameters (weights and biases) are altered to achieve error minimisation. For parameter learning, a technique named gradient descent is used, which calculates a gradient vector, measuring the error's partial derivative in relation to each weight. Consequently, the objective function's minimum is approximated by

altering the weights in the reverse direction of the gradient vector. Hence, the error is reduced. Typically DL models have a large number of weights to adjust [4].

Learning Rate (LR) defines how fast the updates occur during the learning process. Using a small value for this hyperparameter leads to slower convergence, whereas higher values will take less time during the training but might cause fluctuations. To avoid fluctuations some adaptative algorithms like Adadelta, Adagrad and Adam were introduced, that adapt the LR regarding the gradients of the preceding steps [25].

After the training, a testing phase evaluates the model performance with unseen data. This enables the assessment of the generalising capability of the model [4, 24].

To compute the gradient vector, the backpropagation method is used. Backpropagation is based on the chain rule of derivative, meaning that the gradient of an input of a layer can be computed starting with the gradient of its output and moving backwards. This logic is used for all network layers, simplifying the calculus of the gradients regarding the weights of each layer [4].

2.1.3.1 Overview of Common Architectures

The main architectures for Supervised Learning are **Multilayer Perceptron (MLP)**, **Convolutional Neural Networks (CNNs)** and **Recurrent Neural Networks (RNNs)** [2].

Artificial Neural Networks (ANNs) are non-linear computational models aim to the biological systems' capability of processing information [28].

MLPs are a type of feedforward ANN, considered as the basis of DL models [2, 24]. Feedforward means that information flow has one direction without feedback connections, where the model outputs are fed back in the network [24].

The layers of MLPs are composed of neural units or neurons, and each layer can have several units. At each unit, an activation function is used on the weighted sum of their input [2, 3, 4, 28].

The activation functions are based on the way human neurons either fire or not. These introduce the non-linearity between the input-output relationships, transforming the data into a different plane where it is easier to perform classification. Some commonly used are Hyperbolic Tangent (Tanh), Sigmoid, and the Rectified Linear Unit (ReLU) (see Figure 2.3 (B)) [3]. Currently, ReLU, defined by $f(z) = \max(0, z)$, is commonly used over other functions since it learns faster in architectures with multiple layers [4].

The activation function result is fed to the units in the following layer. Hidden layers correspond to the middle layers between the input and output ones [2, 3, 4, 28]. An ANN is considered deep when the hidden layer number is equal or higher than two (see Figure 2.3 (D)), and shallow when it has only one layer (see Figure 2.3 (C)). Using more hidden layers can lead to a more flexible and powerful neural network, capable of modelling intricate input-output relationships [28]. Additionally, MLPs are fully connected networks, having a parameter to describe every input-output unit interaction, which means that every input unit has a connection with every output unit [24].

CNNs are another well-known and frequently used feedforward algorithm in DL [25]. They can process data in the form of multiple arrays, such as images and time-series [4, 24]. This allows

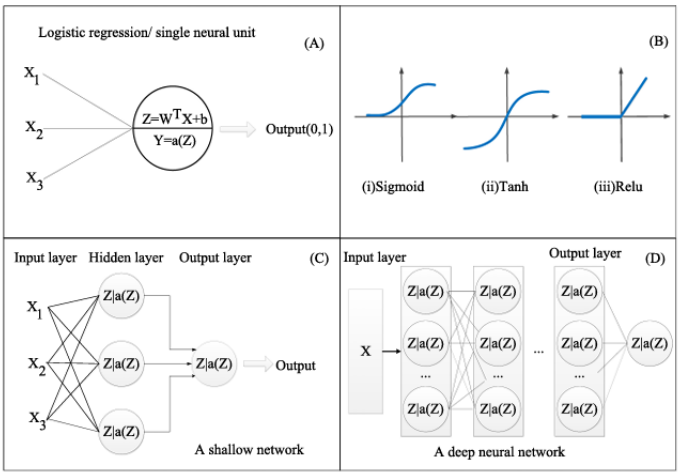


Figure 2.3: Deep and Shallow Neural Network architectures, from [3].

several applications in computer vision (including image recognition and segmentation, medical image analysis), voice data analysis, natural language processing (NLP) and speech recognition [2, 3, 25].

A typical CNN architecture is defined in Figure 2.4. CNNs are composed of different parts: convolutional layers, non-linear activations, pooling layers and finally, fully-connected layers [4, 24, 25].

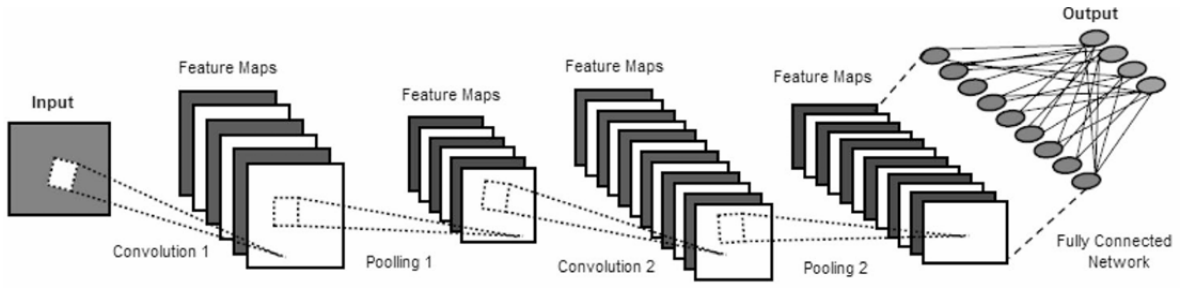


Figure 2.4: CNN Architecture, from [3].

- **Convolutional layers**: aim to identify local combinations of features [4]. This process involves the use of filters or kernels that correspond to a matrix of parameters (weights and biases) and may be considerably smaller than the layer's input. The kernels are convolved with a local part of the input and generate a feature map [4, 25]. As a result, sparse connections are created since the feature map unit is only linked to a small part of the previous layer's feature map [3, 4, 24]. For each convolutional layer, several feature maps are created since different kernels are applied [3, 25]. In a feature map, all the units were obtained through the same filter [4]. An advantage of this parameter sharing is that there are fewer parameters to learn, reducing the complexity of the training and the memory requirements [3, 25]. Then, an activation function is applied to the feature map [4, 25];

- **Pooling Layer**: aims to join the neighbour features with similar meanings through a summary statistic [4, 24]. Some examples are max and average pooling, which take the maximum and the average from a unit's neighbourhood, respectively [24, 25]. This operation reduces the dimensions of the feature map and creates representations that are robust to small variations in the position and to distortions [3, 24]. On the other hand, there is a decrease in the parameter number, that avoids overfitting and optimises the training [25].

- **Fully-Connected (FC) Layers**: identical to MLPs, these transform the data into a higher abstraction level. The Softmax activation function is applied in the last layer to create a vector of probability scores for each classification class [25]. Backpropagation is also used in CNNs to train the network [4].

  Neural networks in DL explore the hierarchical compositions of the data, where the joining of low-level features composes the higher-level features. In images, edges correspond to low-level features that are joined to form motifs, which then form the objects [4]. Using convolutional layers simplifies the identification of the high correlation between local values that form these motifs. On the other hand, it allows the extraction of statistical information invariant to the location of the motifs in the input (equivariance to translation) [4, 24, 25].

RNNs, correspond to specific neural networks that are commonly applied to sequential data applications. They are useful to understand the relations within the data sequence, such as language and speech processing. The sequential data is handled step by step by RNNs. Contrary to CNNs, where there is independence between inputs and outputs, in RNNs, a memory state introduces information about the previous inputs of the sequence in the hidden layers [2, 4, 25]. Figure 2.5, shows the diagram for the RNN on the left and its unfolded version on the right, with the outputs for the hidden layers at different time steps, which can be seen as a feedforward network where all the layers share the same weights. Throughout the entire sequence, the same parameters (U,V,W) are applied at every stage. Backpropagation can be implemented in the unfolded computational graph of RNNs to compute the derivative of the total error regarding the states and the parameters [4].

The training of standard RNNs is difficult since it leads to exploding or vanishing gradients. The network dismisses the influence of the initial inputs as new ones emerge. *Long Short-Term Memory (LSTM)* is an RNN variant that partially resolves the mentioned issues. It introduces memory blocks, composed of memory cells that retain the information at different time periods, in the recurrent connections. Gated units regulate the flow of information, controlling when the data enters the cell states, if this data is kept or removed and defining the output [2, 4, 25].

2.1.3.2 Image Classification Networks

Image classification, a major challenge of computer vision, consists of attributing a class to an image [29].

The success of deep CNNs started in 2012, when the **AlexNet** [30] architecture triumphed in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC), an objection recognition

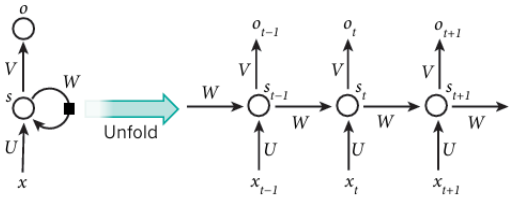
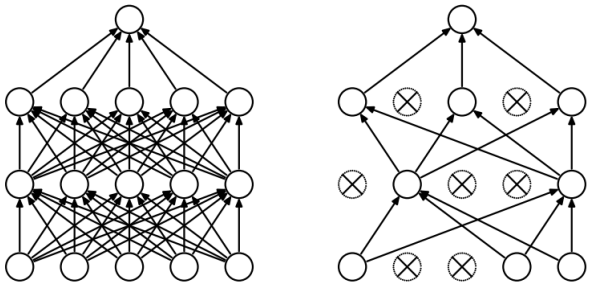


Figure 2.5: Recurrent Networks, from [4].

competition, surpassing previous year's winners by a significant margin. The error rates for the networks proposed for this challenge are presented in Figure 2.7 [8, 15, 29]. It consisted of 5 convolutional layers, some followed by max-pooling, 3 FC layers and a softmax activation function in the output layer (see Figure 2.8). They used large-sized filters (5×5 and 11×11 dimensions) in the first layers. The main problem with the depth increase was the overfitting, which was reduced in AlexNet with dropout [6, 8, 15].

Dropout [5] is a technique that randomly removes some of the connections during the training to prevent the model from excessively adapting to the data. Figure 2.6 exemplifies the difference between a CNN with dropout and without. The probability p defines the probability for each neuron to be used and a value of 0.5 has been considered an appropriate value for different tasks [5].



(a) Standard Neural Net (b) After applying dropout.

Figure 2.6: The effect of using dropout in CNNs, from [5].

Additionally, the introduction of ReLU avoided the vanishing gradient problem, enhancing the rate of convergence. Data augmentation, which corresponds to a set of transformations applied to the dataset images to avoid overfitting, was also applied [6, 8, 15]. With these two techniques and the increase to a depth of 7 feature extraction stages, the learning capability in image classification and recognition tasks was optimised. Hardware was another limiting factor, but AlexNet used two parallel GPUs to overcome this issue [6].

In 2014, **VGGNet** [31] was proposed, with increasing the number of layers to 16-19 (see Figure 2.9). The authors used small-sized kernels (3×3) but doubled the feature maps after the pooling, which allowed them to maintain a receptive field similar to the one produced by larger filters [6, 8, 15]. The complexity was controlled using 1×1 convolutions within the convolutional layers. After the convolutional layer, max-pooling is applied and padding is used to maintain the spatial information. Its main limitation was its considerable computational cost [6].

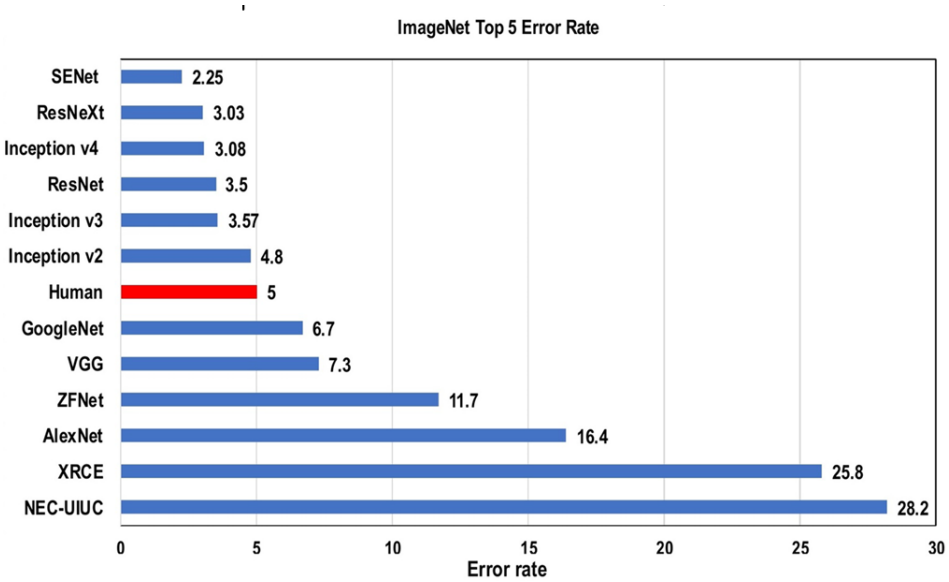


Figure 2.7: Error rates for the different architectures proposed for ImageNet challenge, from [6].

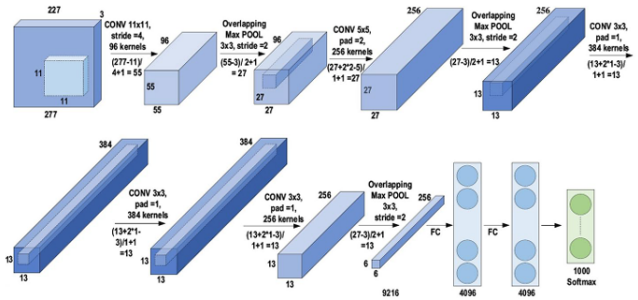


Figure 2.8: AlexNet Architecture, from [6].

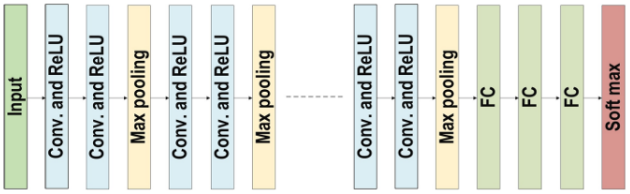


Figure 2.9: VGGNet Architecture, from from [6].

GoogleNet [32] was the winning architecture of the ILSVRC-2014. This architecture has a depth of 22 layers and requires less computational power when compared to the common CNN architectures, while achieving a high accuracy [6, 15]. In this, an "inception module" represented by convolutions is used in parallel with different scales (1×1, 3×3, 5×5) and max-pooling. The concatenation of the feature maps is the next module input (see Figure 2.10). This type of architecture allows to enhance the feature extraction of CNNs [6, 8, 15]. The goal of this architecture was to optimise the computation of the parameters and the learning. The use of a convolutional filter of 1×1 before the other filters regulates the computation. The redundant information is also reduced due to ignoring unimportant channels. Instead of using a FC layer, a global average pooling is introduced, decreasing the density of the connections [6].

ResNet [33], winner of the ILSVRC-2015, was proposed, introducing residual networks, with "short-cut" connections, a bypass pathway concept that allowed to increase the depth of the layers. It was possible to train networks with 34, 50, 101, 152, and even 1202 layers [6, 8, 15]. Since these short-cut connections have a faster convergence, the problem of diminishing gradient is prevented [6].

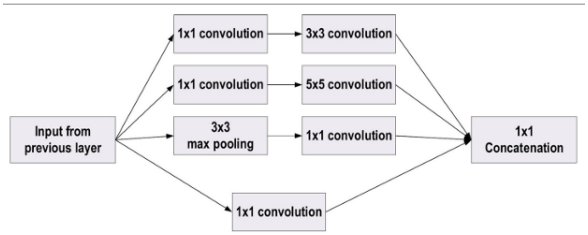


Figure 2.10: Inception Block Architecture Architecture, from [6].

SENet [34] which won the challenge in 2017, is formed of an assemble of Squeeze and Excitation (SE) blocks (see Figure 2.11) and the integration with architectures such as ResNet, improving their discriminative capability [8, 15]. This module works by updating the feature responses in an adaptive way for each channel, considering the relations between the different channels [8].

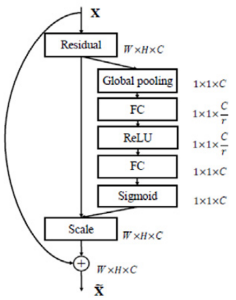


Figure 2.11: SE block architecture, from [6].

2.1.4 Metrics for Classification

Performance metrics allow the comparison between different classification algorithms [7].

Before defining the performance metrics used in this work, we start by defining the following concepts [7]:

- TP: number of outputs correctly predicted as positive when the true value is positive;
- TN: number of outputs correctly predicted as negative when their true value was negative;
- FN: number of outputs wrongly predicted as negative when the true value was positive;
- FP: number of outputs wrongly predicted as positive when the true value was negative.

Precison and **Recall** serve as the foundation for F1-Score and Accuracy. Precision measures the proportion between the correctly estimated positive values and the predicted positive values total. Recall computes the fraction between the correctly estimated positive values and the positive true labels total. In other terms, Precision defines how reliable the prediction of a positive class is, whereas Recall assesses the positive class accuracy [7]. Equations 2.2–2.3 define Precision and Recall, respectively.

$$\text{Precision} = \frac{TP}{TP+FP} \tag{2.2}$$

$$\text{Recall} = \frac{TP}{TP+FN} \tag{2.3}$$

Accuracy defines the overall performance of the model in the dataset by computing the fraction between the number of correct predictions and the total predictions (see Equation 2.4) [7].

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \tag{2.4}$$

F1-Score is the harmonic mean of Precision and Recall (see Equation 2.5). For multi-class cases, Precision and Recall are computed regarding the multiple classes. According to the weight that is given to each class, there are two different metrics: Macro F1-Score and Micro F1-Score [7].

$$\text{F1-Score} = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{2.5}$$

Macro F1-Score (see Equation 2.6) computes the harmonic mean of Macro-Precision and Macro-Recall, that correspond to the mean of Precision and Recall for the individual classes. This approach gives the same weight to each class independently of their class frequency. On the other hand, **Micro F1-Score** computes Precision and Recall without considering the differences between the different classes and its equivalent to Accuracy [7].

$$\text{Macro F1 Score} = \frac{\sum_{i=1}^{n} F1Score_i}{n} \tag{2.6}$$

The **confusion matrix** is a cross table that compares the predicted classes with the real classes, counting the occurrences between them. The diagonal corresponds to the cases when the true values correspond to the predicted ones [7]. Figure [7] shows a confusion matrix example.

		PREDICTED classification				
	Classes	a	b	c	d	Total
ACTUAL classification	a	6	0	1	2	9
	b	3	9	1	1	14
	c	1	0	10	2	13
	d	1	2	1	12	16
	Total	11	11	13	17	**52**

Figure 2.12: Example of a confusion matrix for a multi-classification problem, from [7].

2.1.5 Challenges and Future Directions

Even though current DL methods have shown good results, there are still current challenges this area faces and there are several possible directions for its future [2, 25]:

- **Automatic Label Annotation**: DL highly depends on the amount of data it is trained and its quality. Currently, there is a preference for supervised learning algorithms, that are built with publicly available labelled datasets. However, the quantity of available data has exponentially increased which poses difficulties in the labelling process of these datasets since a manual process is time-consuming and arduous task. The automation of the labelling task can be one of the focus of DL in the future [2, 25];

- **Deepening the investigation of semi-supervised and unsupervised learning**: by adapting existing algorithms or creating new ones might also be another future research focus [2, 25]. The extraction of information from the data can be used to adapt to the emerging data quality problems: heterogeneous, sparse, missing data and noisy data. Therefore, unsupervised learning can be used as a pre-processing method for supervised and semi-supervised methods [25];

- **Black-box perception**: One of the major challenges that hinder the progress of DL is the "black-box" nature of these models as they do not provide any reasoning for the predicted outcomes. Therefore, even though DL models have good performance, they lack statistical interpretability. By contrast, traditional ML models are defined by decision rules that are logic. Understanding the functioning of DL models is crucial for choosing the best approaches for the required task [2, 25];

- **Lightweight Networks**: The performance of current DL models has been related to the use of high computational power, so investigating lightweight networks that can be applied to smart mobiles is also important [2, 25];
- **Emsemble models**: Models that join the characteristics of discriminative and generative models to increase the performance and handle uncertainty in high-risk applications, is a research opportunity [2];
- **Introducing domain knowledge**: Introducing domain-knowledge into the applications areas of DL. An example is in NLP, each language has different rules, introducing the knowledge from this area can help improving results [2].

2.2 Face Recognition

Biometric applications determine the identity of a person through its physiological and behavioural traits [16], such as the face, voice, iris, fingerprint, signature or gait [35]. Face recognition (FR) defines the technology that identifies in videos and images people identities [14]. It is a widely researched computer vision topic [8, 28]. Currently, face has become the most prevalent biometric modality, due to the high performance it has achieved and the advantages it brings over other biometrics, being frequently chosen over fingerprint and iris that are generally considered more precise [14]. Some of these advantages include humans typically using the face as a biometric characteristic for identification [28], making it a natural and easy-to-use technology for identification and authentication applications [15]. Another major advantage of FR is that the user only needs to be in the camera's field of view, being a non-intrusive and user-friendly procedure that does not require the user to participate actively, allowing its application in scenarios where the user does not need to be cooperative such as surveillance [14]. Other examples of applications include border control, home security, smartphone unlock, fraud identification, social media, forensics and law enforcement [14, 15, 16].

The high performance of FR is related to the emergence of DL, the increase of larger databases and more descriptive FR algorithms [36].

2.2.1 Pipeline

An automatic system is usually divided into three building blocks: face detection, face alignment and the FR module (see Figure 2.13) [8].

The face detector detects if a face is present in an image or video, localises it (if present) and returns each face bounding box's coordinates. The bounding boxes should contain the least background possible. This module should be adaptable to changes in illumination, pose and scale [8, 16, 37] .

Secondly, the face landmarks (e.g., eye centres, nose tip) are detected and used to align the face images to normalised canonical coordinates. This step makes FR invariant to the rotation and scaling transformations [8, 37].

Before the FR module, anti-spoofing techniques can be applied to avoid attacks on the FR system. The FR module can be divided into face processing, deep feature extraction and face matching [8].

Face processing is applied to reduce within-subject variability related to the differences in illumination, poses, expressions and the presence of occlusions [8]. The feature extraction step consists of extracting from the normalised faces a discriminative and condensed feature vector (also named template or embedding) that encodes the relevant information that allows the distinction between two different people. Therefore, the templates from the same identity should be similar. Deep networks are trained with a large amount of data guided by loss functions and the trained networks are used to extract the embedding in the testing phase [8, 37, 38].

The face matching algorithm uses a cosine similarity or L2-distance to compute similarity scores between the feature vectors indicating the likelihood of them being of the same identity. This step is crucial since in FR a model is trained to learn a generalised representation rather than being trained to recognise individuals [8, 14, 37].

There are two possible scenarios for face matching: face verification and face identification. For both cases, a gallery with identified individuals is created. In the testing phase, a subject (probe) is compared to one or many images in the gallery. The first case corresponds to face verification, where a 1:1 matching is performed and the objective is to assess whether two images correspond to the same individual. The second defines identification, where the goal is determining the identity present in the image and 1:N matching is performed [8, 28, 38]. Face identification can be a closed-set problem, meaning that the probe subject is included in the dataset or an open-set problem, when it is not certain if the query subject is to be enrolled in the gallery [28, 38].

One of the greatest challenges in FR is to find an adequate metric to compare the facial features [38].

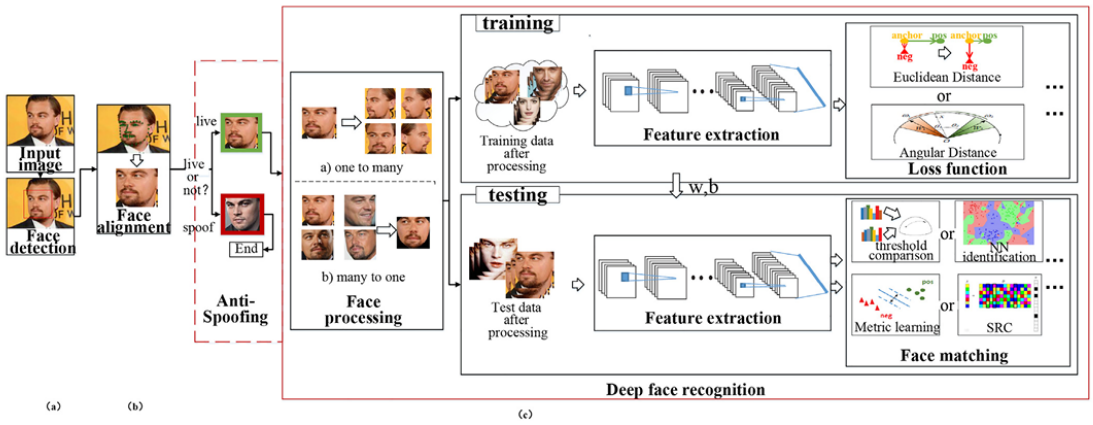


Figure 2.13: Deep FR Pipeline, from [8].

2.2.2 Databases

An essential requirement for training DL FR algorithms is to use large databases [8]. Some of the commonly used datasets for training FR algorithms are:

- **CelebFaces [39]:** Database with 87k images of 5k identities collected from the internet [39]. Later, the same authors proposed CelebFaces+ [40], which increased the number of images to 200k and the number of identities to 10k.
- **MS-Celeb-1M [41]:** Contains 10M celebrity images from 100k identities [41]. It provides several identities but has few images per identity [8].
- **Casia-WebFace [42]:** Contains 0.5M images from 10K identities. It was built with celebrity images extracted from the Internet [8, 15].
- **VGGFace [43]:** Contains 2.6 M images collected from the Internet of 2.6K identities [15].
- **VGGFace2 [44]:** Dataset with 9K identities and 3M images. Since it has several images per identity, there are variations in the pose, illumination and age within the same class. It also includes different ethnicities [8, 15] and provides labels for gender [11].
- **Labelled Faces in the Wild (LFW) [45]:** Contains 13k images, from 5.7k different identities, collected from the Internet. This database was collected to study FR in unconstrained environments [15]. It includes a range of images with different poses, illumination, facial expressions and demographics. It provides labels for 74 face attributes of demographics, data acquisition environment, skin, hair, and accessories. It shares identities with VGGFace2 since both originated with some collected images from the Internet [11].
- **BUPT-Balancedface [46]:** BUPT-Balancedface dataset consists of 1.25M images of 28K different identities, with balance regarding the skin tone. It is balanced by identities, having 7000 identities for each ethnicity [46].

Some of these databases have raised ethical and legal questions due to the methodology used for their collection. Hence, nowadays some of these were withdrawn [36].

2.2.3 Loss Functions

The training and testing identities are separated for the FR to learn a generalised face representation. The training dataset should include the greatest possible number of identities to be able to generalise well. Even though big tech companies like Google and Facebook have databases with 10^6 to 10^7 IDs, the public databases that are accessible to the academic community are limited to 10^3 to 10^5 IDs. Therefore, the focus of the academic community has been to develop discriminative loss functions and architectures that can learn a generalised representation, being an active research area in this field [8, 14].

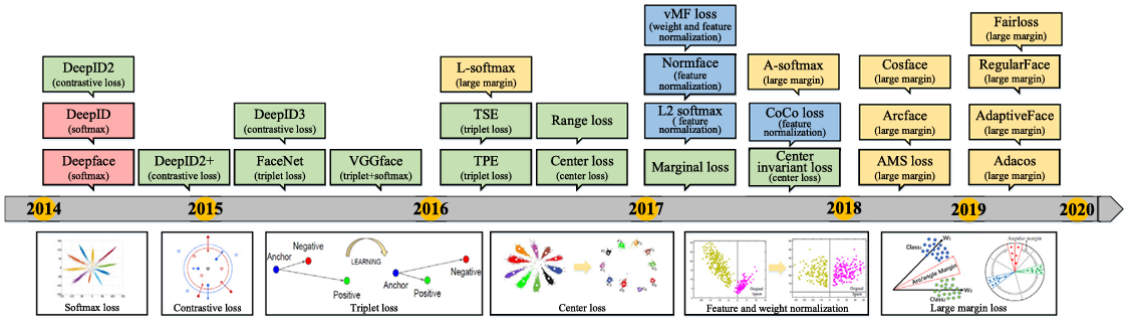


Figure 2.14: Evolution of FR losses, from [8].

In the beginning, the first DL algorithms used the softmax loss (cross-entropy loss with softmax activation layer in the output), following the steps of object classification. However, it proved not to be discriminative enough, achieving inter-class separation but not intra-class compactness [8, 14]. To overcome this challenge, several loss functions have been proposed [14]. Figure 2.14 shows the different proposed losses over the years.

2.2.3.1 Euclidean-distance-based Losses

- **Contrastive Loss:** Reduces the distance between the same person's face templates (positive pairs) and increases the distance between templates from different identities (negative pairs). The main challenge of this approach is the choice of the parameters that determine the positive and negative pairs margins [8].

- **Triplet Loss:** Popular loss in FR. Instead of using the absolute distance between negative and positive pairs, it uses the relative distance between them [8]. More specifically, it uses triplets of the face samples, an anchor, a positive (image with the anchor image identity) and a negative sample (distinct identity from the anchor). The loss ensures that the anchor and the negative sample have a larger distance than a certain margin than the positive sample and the anchor [8, 9, 14].

  Due to the high number of triplets, triplet loss takes more time to converge than softmax loss. To make the convergence faster, hard triplets (triplets that ignore the margin) can be used. Another approach for a faster convergence is to train the model first with softmax and then optimise it with triplet loss [14]. Additionally, training instability can occur in triplet and contrastive loss if the triplets/pairs are poorly chosen [8].

- **Centre Loss:** Defines class centres and introduces a penalty on the distance between the features and their centres [8, 14]. Simpler implementation since there are no triplets or pairs. This loss reduces the intra-class variations and when implemented together with softmax, inter-class variations are also increased [8].

- **Range Loss:** Derived from centre loss [8]. Reduces the k larger distances between the features and their respective class centres. Moreover, it increases the distance between nearest

class centres of each training batch [8, 14]. It can also be combined with softmax loss for better performance.

Instead of learning new losses and combining them with softmax, some authors proposed to adapt softmax loss itself. Some approaches introduced feature normalisation to learn more robust features whereas others focused on introducing an angular or cosine margin [14].

2.2.3.2 Angular/Cosine-margin-based loss

Softmax decision boundary for a binary case (class 0 and 1) is defined as $(W_0 - W_1)x + b0 - b1 = 0$, being W_0,W_1, the weights, b_0,b_1, the biases, and x the feature vector. Several approaches were proposed to reformulate softmax loss to have a more solid decision boundary between samples and to reduce the number of wrong classified samples [8]:

- **Large Margin Softmax (L-Softmax):** The authors added a multiplicative angular margin in the softmax loss between the decision boundary of each class. With the biases equal to zero, assuming a binary classification, the decision boundary is defined in Table 2.1 [8, 14]. This loss is usually joined with softmax due to its difficulty in converging [8].
- **A-Softmax:** Built on L-Softmax work, A-Softmax normalised the weight vectors with L2-norm (unit norm), making the decision relying on the angles only and the weight vectors belong to an hypersphere [8, 14]. In this way, face features belong to a hypersphere manifold and are separated by an angular margin [8]. The decision boundary for this loss is defined in Table 2.1.
- **Additive cosine/angular margin:** These losses do not need to fine-tune complicated hyperparameters as in L-softmax and can be implemented without softmax. Hence, simpler to deploy and optimise than the multiplicative margin [14]. CosFace and AMS loss added the margin to the cosine and normalised both the weight and feature vectors. ArcFace loss improved the geometric interpretation by adding the margin to the angle itself. The decision boundaries for both losses are defined in Table 2.1 [8].

Table 2.1: Decision Boundary for binary case for class 1, with s being the normalised features.

Loss Function	Decision Boundary
Softmax	$(W_0 - W_1)x + b_0 - b_1 = 0$
L-Softmax	$\|\|x\|\|(\|\|W_0\|\|cos(m\theta_0) - \|\|W_1\|\|cos(\theta_1) = 0$
A-Softmax	$\|\|x\|\|(cos(m\theta_0) - cos(\theta_1) = 0$
CosFace	$s(cos(\theta_0) - m - cos(\theta_1))$
ArcFace	$s(cos(\theta_0 + m) - cos(\theta_1))$

2.2.3.3 Variations to softmax loss

Feature normalisation has demonstrated an enlargement in the feature discriminative power [14]. Besides implementing weight normalisation before applying additive margin, some works focused on implementing normalisation by itself [8]. L2-softmax normalised the features to have unit L2-norm [14]. Since the L2-norm is related to the face image quality, this ensures the images have the same attentiveness regardless of their quality [8]. Hasnat et al. [47] normalised the features to ensure a zero mean and variance of one [8, 14]. Ring loss adapted the features to a learned parameter rather than a fixed value [8].

2.2.4 Network architecture

The common FR backbone architectures followed the evolution of deep object classification architectures, namely the state-of-the-art for the ILSVRC, from AlexNET to SENet (see Figure 2.15) and the performance of FR architectures also improved continuously [8, 14].

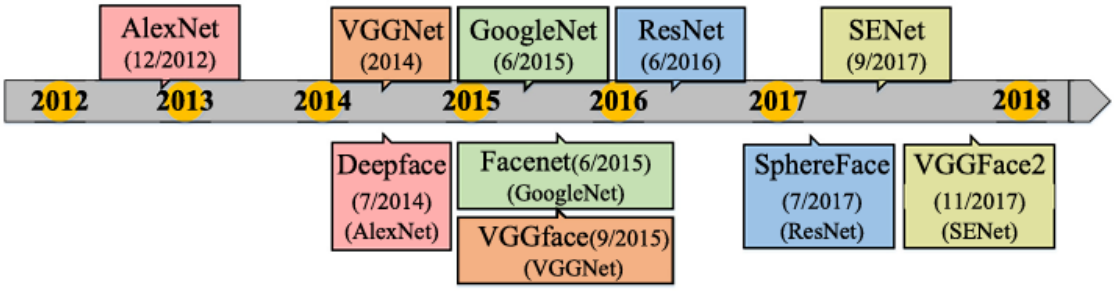


Figure 2.15: Evolution of Object Classification DL Architectures and FR Networks, from [8]. The same colour represents the same architecture.

In 2014, Taigman et al. proposed **DeepFace** [48] that resulted from research on AlexNet architecture [15]. It was one of the earliest CNNs applications for FR, achieving a 97.35% verification accuracy in the LFW [45] benchmark, having a 27% increase in the performance, in comparison with previous state-of-the-art [8, 14, 16]. With the achievement of a near human verification performance [48], there was a shift in the FR investigation to start using DL methods, leading to significant growth in the performance in just three years [15]. It was also one of the first applications to use a 3D model for face alignment [14, 15, 37].

It trained a 9-layer CNN with softmax loss. Contrary to common CNNs, it introduced some locally connected layers that do not share weights, learning different features from different image regions [14, 15, 37]. The network was trained for an identification task, allowing the extraction of robust feature vectors [48]. It was trained with a large dataset of 4.4 million images of around 4000 subjects [14, 15, 16].

Based on the **DeepFace** work, Sun et al. developed **DeepID**. Different versions were introduced over the years, leading to continuous improvement in the performance of models in the dataset [15]:

- **DeepID (2014) [40]:** Used 60 CNNs to extract features from 60 patches (10 different regions, 3 scales and 2 color channels (grey and RGB)) [14, 40]. The aim was to extract features for face verification, however, the CNNs were trained for identification, classifying approximately 10000 identities. Since identification is a more demanding task, the model can learn robust features for verification. The facial features, defined as Deep hidden IDentity features (DeepID), can be extracted from the last hidden layer of the network. The quantity of extracted features decreases along the network, meaning that the final layers have more global features. The introduction of complementary regions to obtain the face features, DeepID allowed an increase in the LFW dataset accuracy (97.45%) with a weak alignment of the faces [15, 40].

- **DeepID2 (2014) [49]:** achieved 99.15% verification accuracy and reduced the error rate by 67% compared to DeepID (which was the best DL solution at the moment) [49]. This approach solved the challenge in FR of achieving good intra-class compactness and great inter-class separability altogether [15, 49]. The features extracted with this algorithm are named Deep IDentification-verification features (DeepID2). This is possible due to using two supervisory signals: identification and verification. The first pushes the feature vectors of different identities apart, whereas the second forces the DeepID of the same person to be similar. Using both signals together increases the results when using them separately since they focus on different feature learning characteristics. The authors used 25 patches selected from the original DeepID [49].

- **DeepID2+ (2014) [50]:** Improves the accuracy on the LFW dataset regarding DeepID2, achieving 99.47% on the LFW dataset [50]. This is due to the use of higher dimension layers, a higher training dataset and the introduction of supervision in the initial convolution layers, which was only introduced in the final layers in DeepID2 [15, 50]. They also use 25 face patches as in the last proposed approach [50]. It surpassed human performance on LFW [51].

- **DeepID3 (2015) [51]:** Ensembled model of two architectures based on VGGNet stacked convolutions and GoogleNet inception layers, increasing the depth of the network when compared to DeepID2+ [15, 51]. Achieves a verification accuracy of 99.53% LFW. It also uses supervised signals for verification and identification [51].

The Google model, **FaceNet**, was proposed in 2015 by Schroff et al. [9]. This CNN approach fine-tunes a 128-dimensional embedding directly through the triplet loss, rather than focusing on optimising the features of a bottleneck architecture as previous DL approaches. The authors use an online triplet mining approach to generate the triplets [9, 15, 37]. In the model architecture (see Figure 2.16) L2 normalisation is applied after the deep CNN to extract the embedding and then the triplet loss is used for training [9]. The model accomplished an accuracy of 99.63%. It was trained on 200M images, corresponding to 8M identities [37]. It explored two architectures:

GoogleNet and the ZF-NET [52], winning the architecture of the ILSVRC challenge in 2013. Both had similar performance, but GoogleNet had a smaller number of parameters [8].

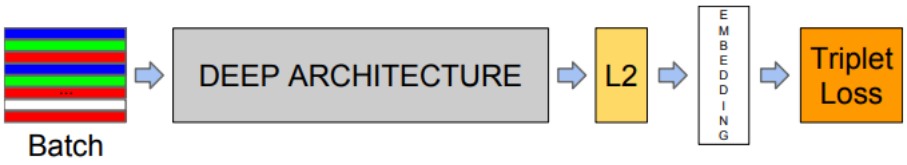


Figure 2.16: FaceNet architecture, from [9].

In 2015 **VGGFace** [43] was also proposed. The authors proposed a massive dataset with images taken from the Internet (VGGFace Dataset). Later, this dataset was used to train the VGGNet model, optimising with a triplet loss and face alignment. With this work, the authors achieved an accuracy of 98.95% on the LFW [8, 15].

Lately, ResNet has become the dominant choice for the backbone in several object recognition tasks, including FR [14]. **SphereFace** [53] in 2017 achieved a 99.42% accuracy in the LFW dataset. The model architecture consisted of a ResNet with 64 layers trained with the A-Softmax loss [8, 15]. The authors of ArcFace [54] studied the trade between accuracy, depth and velocity of ResNet architectures. The model with the best results was the 100-layer ResNet [14].

In late 2017, Cao et al. [44] introduced the **VGGFace2** database, a large dataset with images acquired from Google search. This dataset expresses a wide variety in the profession, ethnicity, pose and age of the collected subjects [8, 44]. The authors achieved state-of-the-art results in the IJB-A and IJB-B datasets, by training a ResNet-50 with SE blocks on the MS-celeb-1M dataset and optimising with their dataset [8].

Table 2.2 summarises the performance, architecture, verification accuracy and type of loss used for the main FR algorithms proposed over the years. Human performance on LFW was assessed by Kumar et al. [55], achieving 99.20% on the original images, 97.53% on cropped images and 94.27% when the focus was on the background, masking out the face.

2.2.5 Challenges and Future Directions

With the great success of FR algorithms, they started being implemented in real-world applications, which brought additional challenges to the academic implementations [8]. Some of the open challenges in FR are:

- **Attacks Against FR:** Some attacks such as presentation, adversarial, template and digital manipulation attacks still present some challenges in the area of FR, so studying methods to make the FR pipeline robust to these attacks is of major importance [8].

- **Data bias:** Some available datasets are composed of images collected from the Internet, very different from those built with images in real settings. Several factors in the data collection process can accentuate the biases, such as the illumination settings, background and

Table 2.2: Performance of FR methods in LFW dataset, adapted from [8]. Acc and Std correspond to accuracy and standard deviation, respectively.

Method	**Year**	**Metric**	**Architecture**	**Training Database**	**Verification Acc±Std(%)**
DeepFace	2014	Softmax	AlexNet	Private	97.35±0.25
DeepID3	2015	Contrastive	VGGNet-10	CelebFaces+	99.53±0.10
FaceNet	2015	Triplet	GoogleNet-24	Private	99.63±0.09
VGGFace	2015	Triplet	VGGNet-16	VGGface	98.95
Centre loss	2016	Center	LeNet+-7	CASIA-WebFace, CACD2000, Celebrity+ (0.7M,17K)	99.28
Range Loss	2016	Range	VGGNet-16	MS-Celeb-1M, CASIA-WebFace	99.52
L-softmax	2016	L-Softmax	VGG-Net18	CASIA-WebFace	98.71
L2-softmax	2017	L2-softmax	ResNet-101	MS-Celeb-1M	99.78
CoColoss	2017	CoCo	-	MS-Celeb-1M	99.86
vMFloss	2017	vMFloss	ResNet-27	MS-Celeb-1M	99.58
MarginalLoss	2017	marginal loss	ResNet-27	MS-Celeb-1M	99.48
SphereFace	2017	A-softmax	ResNet-64	CASIA-WebFace	99.42
AMS loss	2018	AMS	ResNet-20	CASIA-WebFace	99.12
Cosface	2018	Cosface	ResNet-64	CASIA-WebFace	99.33
Arcface	2018	Arcface	ResNet-100	MS-Celeb-1M	99.83
Ring loss	2018	Ringloss	ResNet-64	MS-Celeb-1M	99.50
Human [55]	-	-	-	-	99.20

annotator tendencies. This reduces the model's generalisation capability. Moreover, demographic bias, which includes gender, age, and ethnicity, is present in FR datasets. It has been shown that male, caucasian and middle-aged appear in a larger proportion than other labels. Therefore, mitigating bias and increasing fairness is an important path to follow in FR research. Some approaches are creating new balanced datasets and debiasing algorithms [8]. It is important to mitigate the bias for both the datasets and algorithms [8].

- **"Black-box" nature of FR systems:** Even though FR systems have achieved performances comparable to human performance, they still leave several fundamental questions to be answered, such as the reason why adversarial examples can easily affect an FR system, contrary to humans [8]. The state-of-the-art algorithms for FR are DL models. The increase in performance in this field seems to be connected with a greater lack of opaqueness

of the model due to its higher complexity. Besides adversarial examples, this lack of understanding of the reasoning behind the models can lead to unknown bias, privacy issues and non-transparency. Since it is difficult to directly detect these problems, some possible solutions are: creating models that are interpretable, designing explainability methods for current opaque models and introducing metrics such as fairness in the optimisation of the model [12]. Further research in this area is important to increase public acceptability and reduce the previously mentioned problems.

- **Privacy:** One of the major public concerns in FR is the user's privacy. If the information contained in facial images is leaked, it can lead to some legal and ethical concerns. Therefore, it is important to conduct further research in this area, protecting this information through image/template perturbation, signal mixing and visual cryptography [8].
- **General FR algorithm:** FR has had some successful applications. However, defining a general framework that can be used in different cases is still a difficult task. Even though there is some research on domain adaptation for bias reduction, adapting to new specific domains in a general FR framework is still an open challenge [8].
- **Face Occlusions:** Occlusions can be due to several factors: the presence of accessories: hats, sunglasses, veil; self-occlusions due to the position of the face; related to the low resolution of the images. Occlusions pose two main challenges in FR: the separation between the face feature vectors is increased, increasing the intra-class distance and can lead to alignment errors [15].
- **Heteregenous FR:** Sometimes the FR task needs to be performed with data in different formats: for example, in forensics using drawings to identify suspects, or using infrared images during night, these different techniques should be comparable to RGB images [15].
- **Facial Changes with Ageing:** Accuracy usually decreases when FR is performed across ages due to changes in face characteristics such as skin tone and wrinkles [15].

2.3 Soft-biometrics/Face Attributes Classification

A human face provides more information than just a person's identity, including demographics (e.g., age, ethnicity, gender), accessories, and clothes, which are named face attributes [11]. When describing a person, these characteristics allow a more detailed explanation [56]. FR has focused on using face images for identification and verification, however, due to the broad range of applications, the study of face attributes has gained an increasing interest [11].

Face attributes are also referred as soft-biometrics since they give insights about a person's identity but are not distinct and may not be permanent to fully define it [57].

Face attribute detection can be applied to different tasks: access law enforcement, video surveillance, face retrieval, human-computer interaction, access control and social media (advertising) [11, 58, 56, 59]. They can also provide additional information for identification and

authentication systems, especially in cases where the image quality is low [11, 15, 56]. Some studies have even shown that attributes can be used alone for identification and authentication, being important for privacy-preserving scenarios [11].

The data embedded within face templates is unknown due to the black-box nature of DL FR methodologies. Previous work has demonstrated that social traits, pose, image properties and demographics are present in the templates. In [19] the authors extracted 74 face attributes from the face embeddings. Hence, the study of what information is encoded in face templates is important to develop technologies for reducing bias and for protecting privacy [19] .

There are already some available commercial technologies for face attribute prediction. Nevertheless, these are restricted and the databases used for training are not available to the academic community [58].

Some limitations in face attribute estimation are lighting, occlusions and variations in pose [10, 59]. The study of face attributes as mainly focused on demographics, which can be related to the lack of annotations for several attributes which is a limiting factor [11]. Face attribute prediction is still challenging, especially with unconstrained and uncooperative environments [58].

2.3.1 Databases

- **CelebA** [59]: 200k images of 10k subjects from CelebFaces+. Annotations of 40 binary face attributes. Images with a variety of poses and backgrounds [11, 19];
- **LFWA** [59]: Database with the images from the LFW but labelled with 40 binary face attributes (equal to the ones of CelebA) [58]. 13k images from 6k identities [59];
- **Morph II** [60]: 75k images, 13k identities. Labels for gender (nominal), race (nominal) and age (ordinal) [10, 58].

2.3.2 Multi-task Learning

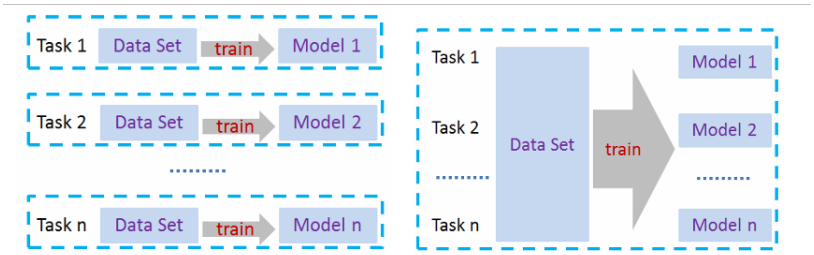
Multi-task learning (MTL) is a learning method that solves several tasks simultaneously, taking advantage of the correlations between the different problems [10, 56]. Usually, MTL methods leads to a better generalisation than single-task learning (STL) ones [10]. Figure 2.17 shows the difference between an MTL and STL framework.

FR has shown to be successful in pose estimation, facial landmark localisation, face detection, and action recognition [56].

2.3.3 Multi-Attribute Estimation

As in other domains, face attribute estimation shifted from hand-crafted features to DL solutions [11, 58].

Initially, research on face attribute estimation focused on predicting a single attribute or using a classifier per attribute when using several attributes. This can bring a heavy computational burden and treat the face characteristics as independent, without considering their correlations



(a) Single-task Learning (b) Multi-task Learning

Figure 2.17: MTL and STL, from [10].

and heterogeneity. To address these limitations, some approaches for simultaneously predicting several attributes emerged [10, 56, 58, 59]. Face attributes are suited for using a MTL framework since other attributes can provide valuable information for estimating that specific attribute due to the strong relations between the attributes [56].

Ehrlich et al. [61] use the face landmarks and PCA-obtained features as input of a MTL restricted Boltzmann machine (MT-RBM) for jointly classifying 40 face attributes of the CelebA database.

Hand et al. [56] proposed an MTL CNN, where two convolutional layers are shared for all the attributes. The authors divided the 40 attributes into 6 categories for the MTL. The shared convolutional layers are followed by 6 branches, one for each category, and each has 2 FC that are then divided in the output nodes for each attribute in the category. Each FC layer is followed by ReLU and a dropout with p=0.5. The authors use the sigmoid loss for training.

Wang et al. [10] the authors proposed an MTL CNN that factors in the attribute correlation and heterogeneity. The network is composed of 5 shared convolutional layers, each followed by max-pooling and 2 FC. These are followed by 2 sub-networks for nominal and ordinal face attributes, that correspond to classification and regression tasks, respectively. For nominal sub-network cross-entropy loss and for the ordinal the Euclidean-distance is used. For classification, softmax is used. In the case of the CelebA, since there are only binary attributes, it only uses the nominal sub-network.

Han et al. [58] proposed a deep MTL network composed of shared layers that are followed by subnetworks for category-specific learning of heterogeneous attributes regarding their semantic meaning (whole or local face attributes) and the type of data (ordinal or nominal). The authors use a network that is based on AlexNet architecture for shared feature learning, using 5 convolutional layers, each followed by batch normalisation (BatchNorm) and a pooling layer. After the convolutional layers, for the feature learning, there are 2 FC. The category-specific layers are composed of 2 FC. Cross-entropy loss is used for the nominal attributes, whereas Euclidean loss is used for the ordinal attributes. For CelebA, several binary attributes are jointly estimated. The network is trained with stochastic gradient descent and weight decay in an end-to-end approach.

Terhörst et al. [19] used a Massive Attribute Classifier (MAC) for estimating simultaneously

several face attributes, using the common feature space through this MTL approach. The authors used prediction reliabilities to classify the face attributes into three predictability categories: easy-predictable, predictable and hardly-predictable. The researchers showed that of a total of 113 attributes, 74 were at least predictable, showing that they were encoded in the face templates. Unexpectedly, variable characteristics such as hairstyles, accessories, and makeup, were the ones that showed results for the easy-predictable category. The FaceNet and ArcFace models pre-trained with the MS-Celeb-1M [41] database were used to extract 128 and 512-sized embeddings, respectively. The authors use the LFW and CelebA datasets for training, splitting the data in a 70% for training and 30% split. The performance is assessed for the whole dataset and for predictions with 50% of the highest reliabilities [19].

Table 2.3 summarises the mentioned approaches and the performance for CelebA, LFWA and MORPH II datasets. For the age prediction in MORPH II, the accuracy corresponds to the accuracy with a 5-year absolute error.

Terhörst et al. [11] proposed a automatic labelling pipeline that transfers the annotations from source datasets (CelebA and LFW) to a target dataset (VGGFace2). This addresses one of the challenges in FR mentioned in the Subsection 2.1.5 of this chapter.

2.3.3.1 MAAD-Face Annotation Pipeline

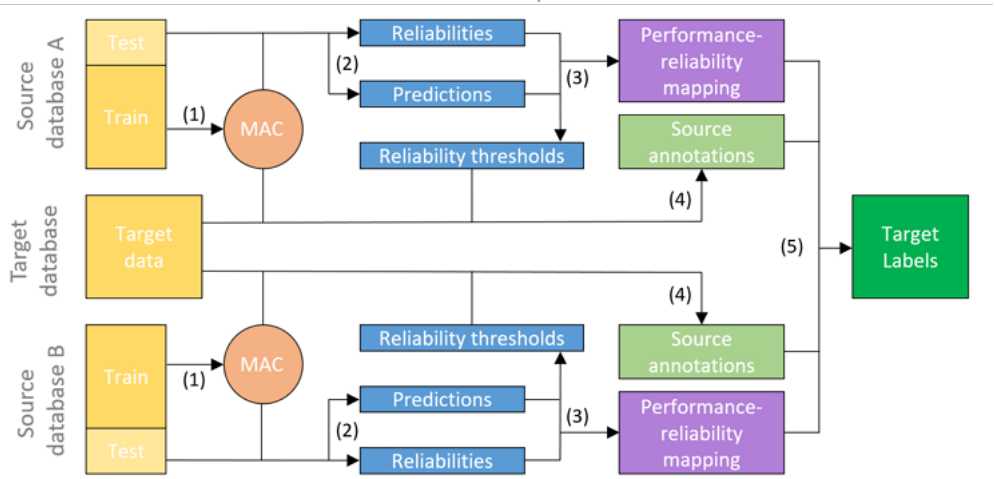


Figure 2.18: MAADFace annotation pipeline, from [11].

This pipeline is represented in Figure 2.18. The authors used the MAC proposed in their previous work [19] and extracted 128-sized embeddings with FaceNet. The authors divided the source datasets into 80% training and 20% testing. The training parts of the source datasets are used to train the MAC. Next, the predictions and their reliabilities are obtained for the test sets. This allows computing a reliability threshold for every attribute, which will then be used to define the low-quality labels as **0** (**undefined**), meaning that the model is not confident enough about these predictions. **True** predictions correspond to the value **1** and the **False** ones to **-1**. Additionally, the reliabilities and predictions from the test set of the source databases are also used to determine the

Table 2.3: Summary of Methods for Multi-Attribute Estimation. Avg is an abbreviation for average.

Publication	Year	Architecture	Database	Avg performance per attribute
Ehrlich et al. [61]	2015	MT-RBM	CelebA	Accuracy: CelebA 87%
Hand et al. [56]	2016	2 common convolutional layers and 6 branched convolutional layers. Followed by 2 FC layers. An auxiliary FC Network is used for enhancing the learning of the attribute relations.	CelebA, LFWA	Accuracy: CelebA 91%, LFWA 86%
Wang et al. [10]	2017	AlexNet for common feature learning, followed by 2 branches for ordinal and nominal attributes, respectively. A head for each attribute of the sub-networks.	CelebA, MORPH II	Accuracy: CelebA 92%. MORPH II: Age: 85%, Gender: 98%, Ethnicity: 97%
Han et al. [58]	2018	5 common Convolutional Layers, followed by BatchNorm and max-pooling layers. After, there are 2 FC layers. Each category-specific layer contains 2 FC layers [58].	CelebA, LFWA, MORPH II	Accuracy: CelebA 93%; LFWA 86%. Morph II: Age: 85%; Gender: 98%; Ethnicity: 99%
Terhörst et al. [19]	2020	Embedding extraction with FaceNet or ArcFace. 1 common FC with size 512. 1 FC layer for each attribute with size 512.	CelebA, LFW	Balanced accuracy: FaceNet: CelebA 83%, LFW 82%; ArcFace: CelebA 80%, LFW 67%.

performance-reliability mapping, which allows mapping an attribute-reliability to its correctness. Thus, the target data annotations are computed with the models trained for both source datasets and if these are different, the one with the highest correctness will be implemented [11].

With this pipeline, the authors created the Massively Annotated Attribute Dataset (MAAD) that contains 3.3M images, corresponding to 9k identities, with a total of 123.9M labels of 47 face attributes [11]. Figure 2.19 shows an example of an annotation from the database.

Neto et al. [36] used this pipeline to label four datasets, two synthetic and two real, and studied the differences between the attribute distribution for the synthetic and real data.



Male	1	Bangs	-1	Round Face	-1	Big Lips	0
Young	0	Sideburns	-1	Double Chin	-1	Big Nose	0
Middle Aged	1	Black Hair	1	High Cheekbones	-1	Pointy Nose	1
Senior	-1	Blond Hair	-1	Chubby	-1	Heavy Makeup	-1
Asian	-1	Brown Hair	0	Obstructed Forehead	0	Wearing Hat	-1
White	1	Gray Hair	-1	Fully Visible Forehead	0	Wearing Earrings	-1
Black	-1	No Beard	0	Brown Eyes	0	Wearing Necktie	1
Rosy Cheeks	-1	Mustache	0	Bags Under Eyes	1	Wearing Lipstick	-1
Shiny Skin	0	5 o Clock Shadow	0	Bushy Eyebrows	1	No Eyewear	1
Bald	-1	Goatee	0	Arched Eyebrows	-1	Eyeglasses	-1
Wavy Hair	-1	Oval Face	0	Mouth Closed	-1	Attractive	-1
Receding Hairline	0	Square Face	1	Smiling	1		

Figure 2.19: Annotation Example from MAAD-face database, from [11].

2.4 Explainable Artificial Intelligence

2.4.1 Introduction

The emergence of DL has led to a performance increase in different tasks [62]. However, DL models are more opaque than traditional ML models, due to the vast amount of parameters that are used for learning, leading to a trade-off between an ML model's accuracy and its transparency [12, 20, 21].

There are already several AI implementations in our daily lives. However, the system's opacity can be dangerous, especially in high-risk areas such as biometrics, law, healthcare, finance, defence, and autonomous driving, where it is necessary to explain the reason for the model predictions since a wrong prediction can lead to a drastic outcome [12, 20, 21, 22, 62]. In these areas there is a contrast between research and their market implementation due to heavy regulation [20]. Additionally, there is usually a lack of trust in systems that are not trustworthy, controllable and interpretable [12, 20].

With the high number of applications and research related to ML and the concerns about the lack of transparency, explainability is becoming increasingly important for different AI stakeholders [12, 20, 21, 62]. Therefore, for ethical, social and legal reasons, there is a high demand for explainable techniques to be introduced in ML systems [21]. The focus has shifted from accuracy to explainability [62]. Given the growing focus on this area, the number of publications in this field has increased [21, 62, 63].

The field of xAI aims to increase the explainability and transparency of AI models while maintaining high performance, creating user trust and understanding and allowing ML success in its diverse applications [20, 21, 22].

Interpretability improves ML systems implementation since it allows other desired goals, such as:

- *Fairness:* xAI can help detect and remove bias from models by understanding which variables are producing the final result, ensuring there is no discrimination against more fragile demographic groups [20, 22, 63].

- *Privacy:* The complexity of the model prevents the knowledge of its inner workings and the captured information. Explainability can explain which information is being stored and if confidential data is safeguarded. Nevertheless, if non-permitted individuals unveil the functioning of the model, the data privacy can be comprised [20, 22, 63].
- *Reliability and Robustness:* Creating a model that is robust to adversarial attacks and reliable [20].
- *Causality:* Interpretable models make assessing the causality between the input and the predicted output easier, either by providing an initial insight or by corroborating already existent causality-inference techniques [20].
- *Utility:* If the model provides useful information for the user to complete a certain task [63].
- *Trustworthiness:* If the user trusts that the model will perform the desired way for a certain task [20, 63].
- *Transferability:* Knowing the model decisions simplifies applying this model to different applications [20].
- *Accessibility/Interactivity:* Explainable models increase the interaction between the model and make it simple [20].

Nevertheless, explainability is not always required [21, 22, 63]. Indeed, introducing explainability in AI models increases the cost and can reduce models' efficiency. The model's ability to deal with errors and its transparency should be taken into account for analysing the need of explainability [21]. There are two main scenarios where an exceptional performance is sufficient. First, in low-risk applications, where a wrong decision will not have a high impact. Secondly, in well-established problems, already tested in real-world scenarios that provide confidence [22, 63].

Doshi-Velez and Kim [63] define that interpretability is required if there is incompleteness in the problem formulation that can lead to unquantified bias. This means that using performance alone is insufficient, leading to a need to provide an explanation [20, 22]. Explanations can be needed in different scenarios [22, 63]:

- To create a scientific understanding;
- To secure safety in complex systems that cannot be tested in every case scenario;
- For ethical reasons to avoid discrimination, detect bias and introduce fairness in the system;
- To ensure the system is optimising the end goal and not an intermediate objective;
- To verify the trade-off between different goals.

Adadi e Berrada [21] distinguish four different reasons for the need for xAI in systems: for justifying the decisions, for having better control over the model, for continuously improving the model and to gain new knowledge.

2.4.2 Terminology

Since xAI is an emerging scientific field, its terminology is still inconsistent [12]. There is no unanimity for defining *interpretability* and how to define a standard evaluation for it [20, 63]. In some literature, interpretability and explainability are used as synonyms [12, 20, 22]. However, interpretability alone does not characterise all the problems associated with the opaque models, being the formalization of explainability important [62]. Even in the literature that separates both concepts, their definition is not consistent. Nevertheless, normally the authors refer to these concepts as interconnected [12, 21].

Adadi e Berrada [21] showed that the ML community has a preference using the term interpretability compared to explainability despite the term being present in the xAI definition.

Neto et al. [12] define these concepts as:

- *Interpretability:* addresses "how" the model works by explaining its inner mechanisms and the logic behind the computed output comprehensibly.
- *Explainability:* tackles "why" the model produced that prediction by providing an explanation for it.

According to the authors, it is possible to deduct the "why" from knowing the functioning of the model, but the opposite is not verified [12].

Arrieta et al. [20] introduce the end-user as a conditional factor of explainability, since the perception of the explanation depends on the knowledge of the topic of the person.

Some authors also define *transparency* as the capacity of models being inherently understandable, being the contrary of "black-box" [20, 62]. Decision trees, rule-base learning, k-nearest neighbours, and bayesian networks are considered transparent models [20, 62]. A model being transparent does not necessarily imply it is interpretable [62]. According to [20] it depends on the end-user.

Other methods such as support vector machines, random forest and DL models show great performance but lack transparency [62].

2.4.3 Taxonomy

There is also no full agreement on the categorisation of explainable AI methods [12].

The taxonomy of xAI methods can be divided according to its coverage (local or global), usage (post-hoc or not) and its methodology (feature-based or parameter-based) [62].

Some researchers divide them into pre-, in- and post-models, others just if they are post-hoc or not [12]:

- **Pre-model methods** prioritise data knowledge and understanding, enhancing trust in the model's decisions and subsequent explanations. Visualisation methods are a common approach for this purpose [12].

- **In-model methods** introduce the interpretability in the model itself through the use of some condition. These include types of ML that are intrinsically interpretable and constraints, such as regularisation terms and constraints for achieving a specific property (e.g., monotonicity). These are limited by the model properties and feature properties [12].
- **Post-model or Post-hoc methods** produce explanations for the outputs computed by a model. A widely used method involves adding perturbations to the input and evaluating how these changes affect the model output [12]. Some examples of post-model methods include [20]:
  - Text explanations: Text-based explanations that explain the model's predictions;
  - Visualisations: Give a visual explanation of the functioning of the model;
  - Local explanations: Focus on explaining relevant parts of the solution;
  - Explanations by example: Provide significant examples to illustrate the model behaviour;
  - Explanations by simplification: A simpler model is created from the original that still maintain its performance;
  - Feature relevance: Methods that provide a relevance score for the model variables.

Arrieta et al. [20] divide the models into transparent or post-hoc. Transparent models are considered with three levels of transparency: *algorithmic transparency*: if the user can understand the reasons behind the model predictions, *decomposibility*: if each segment of the model can be explained; and *simulatability*: if a human is capable of simulating it.

Other differentiate xAI approaches into:

- **Model-Agnostic:** Methods that can be adapted to different model classes without depending on a specific model architecture [20, 62];
- **Model-Specific:** Methods designed specifically for a certain type of model architecture [20, 62].

Generally, in-model approaches are model-specific and post-model are model-agnostic, but this is not always verified [12].

Post-model/Post-hoc approaches have been the main focus in this area and usually the type of approach that is associated with explainability. This can be related to the less engineering effort it needs and the high requirements of the performance metrics that are required, being the best trade-off between the performance metrics and effort needed. They are more convenient [12].

Neto et al. [12] propose a new taxonomy for xAI methods motivated by the relation between causality and xAI. Causality uses mathematics to describe the causal relationships that determine the succession of events. Similarly, xAI has the goal of explaining the relationships between the input and the output. The authors believe that current classifications do not include approaches that focus on understanding the model itself. Hence, they propose a xAI ladder, based on the

causality ladder [64] (see Figure 2.20). Within this framework, the goal is to find a causal system that represents the causal relations of the model.

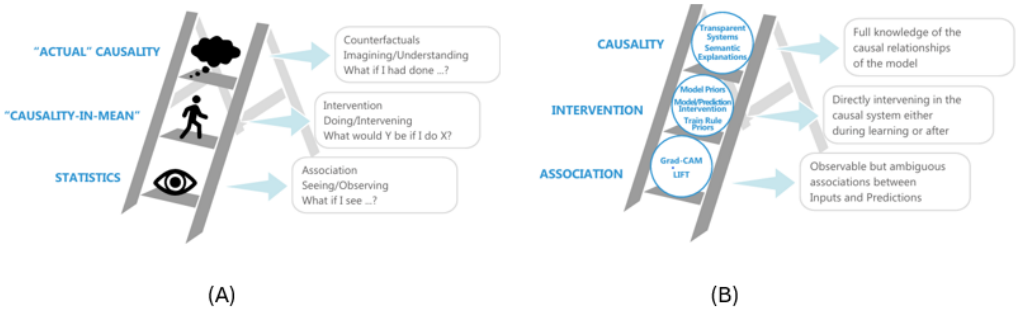


Figure 2.20: A- Ladder of causality, B- xAI ladder, from [12].

This method allows the quantification of the causality level at each different step and also it facilitates the detection of wrongfully learned relations [12].

The proposed xAI framework can be divided into three steps. Nevertheless, the connection between the different steps is continuous, since some methods are between two different categories:

- **First step**: The most basic level of explanations. Correspond to observational explanations that include both pre-model and post-model approaches. These allow the understanding of the location of the features that contributed to the final output, but it does not provide any detail regarding why these were important for the decision. The analysis of such methods depends on the person assessing them. Besides, there is important information missing from these explanations [12].

- **Second step**: Include post-model and in-model methods. Post-hoc approaches can be by an intervention on the input (understanding the causal relations through input modification) or by an intervention in the prediction (by analysing the inputs of similar predictions). In-model methods use training constraints or priors applied to the models creating a predicted behaviour. This allows the partial understanding of the causal system that represents our model. Several approaches fit this ladder and some can be used to build a naive counterfactual [12].

- **Last step**: Includes methods that completely explain the underlying causal relationships within the deep model. This would allow the answering of the "How?" question and consequently, the "Why" and "Where". An agreement between xAI scientists is that counterfactuals are an effective way to make models understandable for humans. The methods in this ladder allow the creation of counterfactuals that entirely explain the model's functioning, whereas the second ladder only allows the creation of naive counterfactuals that are based on searches of the latent space. This corresponds to the highest level of explanation [12].

Current approaches for xAI are mostly located between the first and the second ladder, being able to locate what are the variables that contribute the most for the decision, but not fully answering the question "Why?". Nevertheless, not knowing the inner mechanisms of the model leads to security problems and a lack of user trust [12].

2.4.4 Evaluation of Interpretability

Common metrics should be established to compare and support existing xAI methods. These should allow the quantification of the model performance regarding certain explainability factors [20].

Doshi-Velez and Kim [63] propose for the evaluation of interpretability 3 categories:

- **Application-grounded:** The explainable model is assessed within its real context by the target user. This is the most effective way to check whether the model can perform the desired task. Explanations produced by humans can be used as a baseline;
- **Human-grounded:** Human-based evaluation of tasks that simplify the desired application but still maintain its substance. This is useful when conducting experiments with the domain experts is difficult. Hence, it is cheaper and its easier to find subjects to perform it since domain experts are not required;
- **Functionally-grounded:** The quality of the explanation is assessed through a formal definition of interpretability. It is performed without human intervention, which leads to a reduced cost and time when compared to the other categories. Commonly used if certain model classes have already been subject to human-based evaluation.

2.4.5 Shapley Additive Explanation

SHapley Additive exPlanation (SHAP) [65] is a method used for explaining ML models that originated from the shapley values definition from game theory [62]. The shapley values measure the contribution of each player in a coalition game. In ML, these values measure the importance of each feature for a certain prediction [62, 65].

The SHAP value, ϕ_i , of an input feature, i, can be defined as [13]:

$$\phi_i = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!(|M| - |S| - 1)!}{|M|!} [f_x(S \cup \{i\}) - f_x(S)] \tag{2.7}$$

Where M is the number of input features, N corresponds to a set of features, S correspond to a subset of the set of features without the feature for which the SHAP value is computed, $f_x(S)$ corresponds to the value of the output for the subset when all the features are included and $f_x(S \cup \{i\})$ to the prediction when the feature i is removed [13].

Hence, SHAP values are weighted average marginal contribution of a certain feature for every possible feature combination [62].

The exact computation of SHAP values is complicated but can be approximated using the knowledge from already existent additive feature attribution approaches. There are approximation methods that are model-agnostic (e.g., Shapley sampling values, KernelSHAP) and others that are model-specific (e.g., Linear SHAP, Low-Order SHAP, MaxSHAP, DeepSHAP) [65].

SHAP summary plots use the feature's individual contributions to plot a summarised representation of the feature's importance, the distribution and scale of the feature's impact on the prediction and the association between the value of the feature and its impact. The features are ranked by their global importance. The graphic has blue and red color that corresponds to a low or high value of the feature. Figure 2.21 demonstrates an example for this type of plot [13].

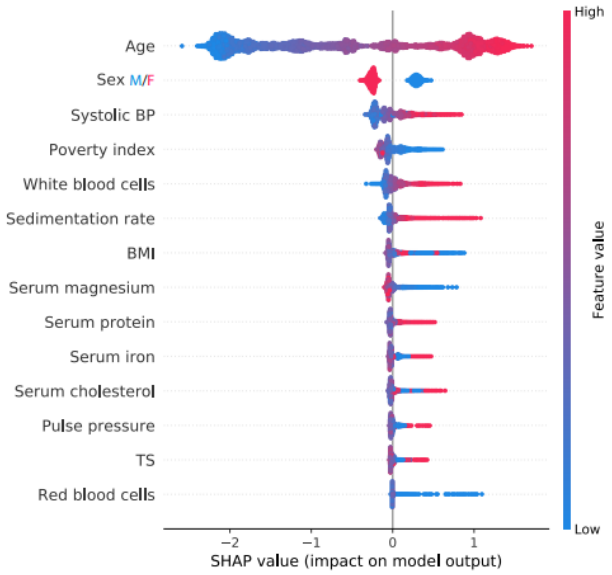


Figure 2.21: SHAP summary plot example, from [13].

These can be considered as feature relevance and visual explanations (when they are presented in a graphic format).

2.4.6 Saliency Maps

Saliency maps are a common approach used in computer vision. Saliency methods define the feature importance for the model output and this can be visualised through saliency maps that provide different colour intensities for the different feature importances in the input [17, 18, 23].

Saliency methods can be divided into two common types: perturbation-based and backpropagation-based [17].

Pertubation-based methods implement some change in the input and assess the impact of that change in the model's predictions. These can require a high computational cost but are model-agnostic. One example is Randomized Input for Explanation (RISE) [66]: This method assesses the differences in the model output when different binary masks are implemented on the input.

The probabilities are used for saliency maps, allowing the visualisation of each pixel's importance for the model's output [17, 66]. SHAP values can also be introduced in this category since they can be computed for each pixel in the input image [23].

Backpropagation-based methods make a backward pass of a relevance score from the output to the input of a trained model, allowing the identification of the most relevant features for that input [17]. These types of approaches are usually model-specific. One approach is Excitation Backprop (EBP) [67] that measures the conditional probabilities of the most important neurons for the prediction when passing a signal from the output to the input [23, 67]. Contrastive EBP (c-EBP) and Truncated Contrastive EBP (tcEBP) are two adaptations of this approach that can be used for comparing image pairs [23]. Other approaches within this category are gradient-based. These include class activation mapping (CAM) [68], these methods use global average pooling layers rather than FC layers in CNN architectures. The saliency of each class is obtained through the weighted sum of the different feature maps for each class. Gradient Class Activation Mapping (Grad-CAM) [69] is an extension of this method where the feature maps are weighted by the gradient of each class regarding each feature map [17]. Sundararajan et al. [70] introduced an approach named integrated gradients, where the gradients for the same dimension of a reference image and an image sample are integrated and used to obtain a saliency map [23].

2.4.7 Explainability in Face Recognition

Similar to other fields, the increase in performance in Biometrics models was associated with its greater opaqueness. Wrong decisions made by these systems can create undesirable situations, such as discrimination. Additionally, the lack of transparency can make these systems susceptible to different types of attacks. Therefore, there is a need to introduce explainability and interpretability in these systems [12].

Several efforts have been made to increase the user's confidence in these systems. However, it has been shown by different authors differences in performance across different demographic groups. The bias present in biometric systems is a limiting factor in its progress. Some authors believe these are unfit for market implementations [12].

The introduction of explainability in biometric systems increases users' trust and allows the detection of attacks, facilitating the detection of bias and anomalies in the functioning of the model [12].

In the past few years, there has been an increase in the number of xAI approaches proposed for biometrics. Nevertheless, xAI is not as developed in biometrics as in other fields and many of these approaches have still not been applied or properly adapted to biometrics. The approaches for FR are mainly focused on post-model methods since in-model approaches can affect the model's performance. Particularly, the focus has been on visual approaches, interventions in the input and output and the use of model priors and training rule priors [12].

Face is the most common modality for explainable biometrics. Different areas of face biometrics have explored interpretability and explainability methods, such as FR, face morphing, face expression/emotion and presentation attack detection [12].

xFR is the research area that focuses on explaining the reason behind matching or not two faces [17, 18]. Interpretability and explainability are specially required in security applications [18].

Similar to other computer vision areas, in FR, a common approach includes the use of visual explanations [12, 18, 23], being saliency maps one of the focus in this area [18, 23]. However, general saliency map methods applied to other domains (e.g., Grad-CAM), cannot be directly applied to FR since these are optimised for classification tasks and FR focuses on extracting a face template. On the other hand, the difference in the performance can be related to the structure of the face [17, 18, 23].

Lu and Ebrahimi [17] introduced an approach for xFR named Similarity-based RISE (S-RISE) that adapted the RISE algorithm for the problem of face verification. In this method, the authors generate saliency maps for pairs of images that can be for matching $\{I_p, I_m\}$ or non-matching pairs $\{I_p, I_n\}$. For each image pair, several masks are generated. For each mask, using the example of the matching pair, a masked I_p is introduced in the FR system with the unmasked I_m and the unmasked I_p with a masked I_m in the face verification system and the similarity scores for that mask for each image are computed. The weight for each mask for the image will be the similarity score. The saliency map for that image will correspond to the weighted sum of the masks.

Other proposed methods for xFR changed the models' training through priors, regularisations and constraints [12]. Williford et al. [71] introduced an approach that uses both training rule priors, intervention in the input and the prediction [12]. The authors used two different approaches: subtree EBP and Density-based Input Sampling for Explanation (DISE). The first corresponds to an adaptation of EBP, cEBP and tcEBP for face triplets that replace cross-entropy loss for triplet loss [17]. The latter is an approach that adapts the RISE algorithm for being implemented on face images. The authors use prior density for generating the binary masks [71].

Huber et al. [18] introduced Similarity Score Argument Backpropagation (xSSAB). This approach used a siamese network that passed two images through an FR extraction module to obtain the two face templates and then used an additional layer that measured the cosine similarity between these two embeddings. Maps of the similarity and dissimilarity of the two face images were generated through the backpropagation of gradients, that are computed based on the decision threshold and the similarity scores.

Chapter 3

Face Attributes Classification

3.1 Methodology

Figure 3.1 shows a detailed version of the pipeline presented in Chapter 1. As mentioned in that Chapter, the final goal was to build an explanation based on the location of the face soft-biometrics. Hence, the first step was to train a face attribute classifier using a dataset labelled with face attributes. Due to the correlation between the labels of the used dataset, an MTL approach was followed.

Two distinct embedding sizes were available, 128 and 512 dimensions. First, different experiments were conducted to determine the best architecture for this classifier, including ablation studies and hyperparameter tuning. For these studies, the classifier was trained with the 128D embeddings to predict gender and ethnicity. Secondly, the performance of the classifier trained with the two different embedding sizes was compared. Finally, seven more attributes related to face, mouth and nose shape were included as additional attributes to be predicted by the multi-task model. A comparison of the different embedding sizes for this model was also implemented.

Therefore, results for four different models that share the same architecture are reported but change either the dimension of the input or the number of attributes that are being classified. **MTL1** and **MTL2** correspond to the architectures implemented for the two or nine attributes, respectively. **A** or **B** corresponds to a 128 and 512 input dimension, respectively.

Finally, an explanation approach based on SHAP values was implemented to analyse the feature contribution to the final output decision. This part is discussed in Chapter 4.

3.1.1 Dataset

The dataset BUPT-Balancedface [46] and the available annotations for 47 face attributes [36] were used. This dataset originally included labels for ethnicity, the 47 labels were produced by the automatic annotation pipeline proposed in [11], previously mentioned in Section 2.3. The same source datasets (LFWA and CelebA) were used to generate the labels for the BUPT-Balancedface.

Figure 3.2 shows the correlation between the 47 attributes in the annotations for the dataset. As expected, facial hair is heavily correlated with being male. On the contrary, attributes such as

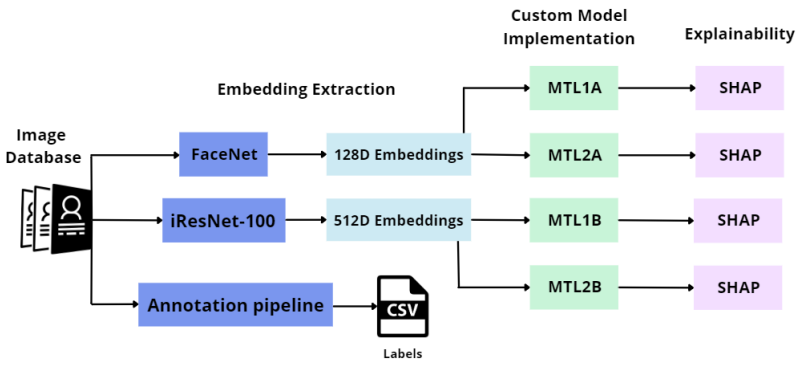


Figure 3.1: Pipeline for the developed work.

being attractive and related to wearing makeup are negatively correlated to being male. The latter can be somewhat controversial, so labels such as *attractive* should be removed. Some of these attributes are subjective and can depend on societal concepts, thus being less interesting for this study.

As mentioned in Section 2.2, this database is balanced for ethnicity. Figure 3.3 shows the ethnicity label distribution from the dataset.

From the 47 available labels, only 8 were used for the following experiences. For the first approach, the most common demographics (i.e., gender and ethnicity) were selected. Rather than using the ethnicity labels available from this 47, the ones already provided by the original dataset were used. The identities in the dataset were separated according to the ethnicity: Asian, African, Caucasian and Indian.

The seven other labels were selected based on their permanency since the goal was to create an explanation approach. This included labels for face geometry, such as face shape (square, round and oval face) and high cheekbones, nose type (pointy and big nose) and mouth shape (big lips). These are labelled in 0: undefined, 1: positive and -1: negative. The undefined labels correspond to a low confidence in their prediction from model used for the automatic annotation pipeline [11]. Figure 3.4 shows the distribution of 8 labels from the annotations included in the described work.

Additionally, it is possible to see in Figure 3.5 that the selected attributes for classification also have positive and negative relations between them, which makes them suitable for an MTL framework.

As mentioned above, rather than using the original images, face templates were used as the input for the classifiers. Two different dimensions were used to assess the different performance. The 128 and 512-dimension embeddings were extracted with *FaceNet* and *iResNet-100*, respectively.

Each embedding dataset was separated into a train, validation and test set in a split of 80%-10%-10% regarding the number of identities, avoiding the presence of common identities in the different splits. Hence, 22.5k identities were available for training, 2800 for validation and 2800

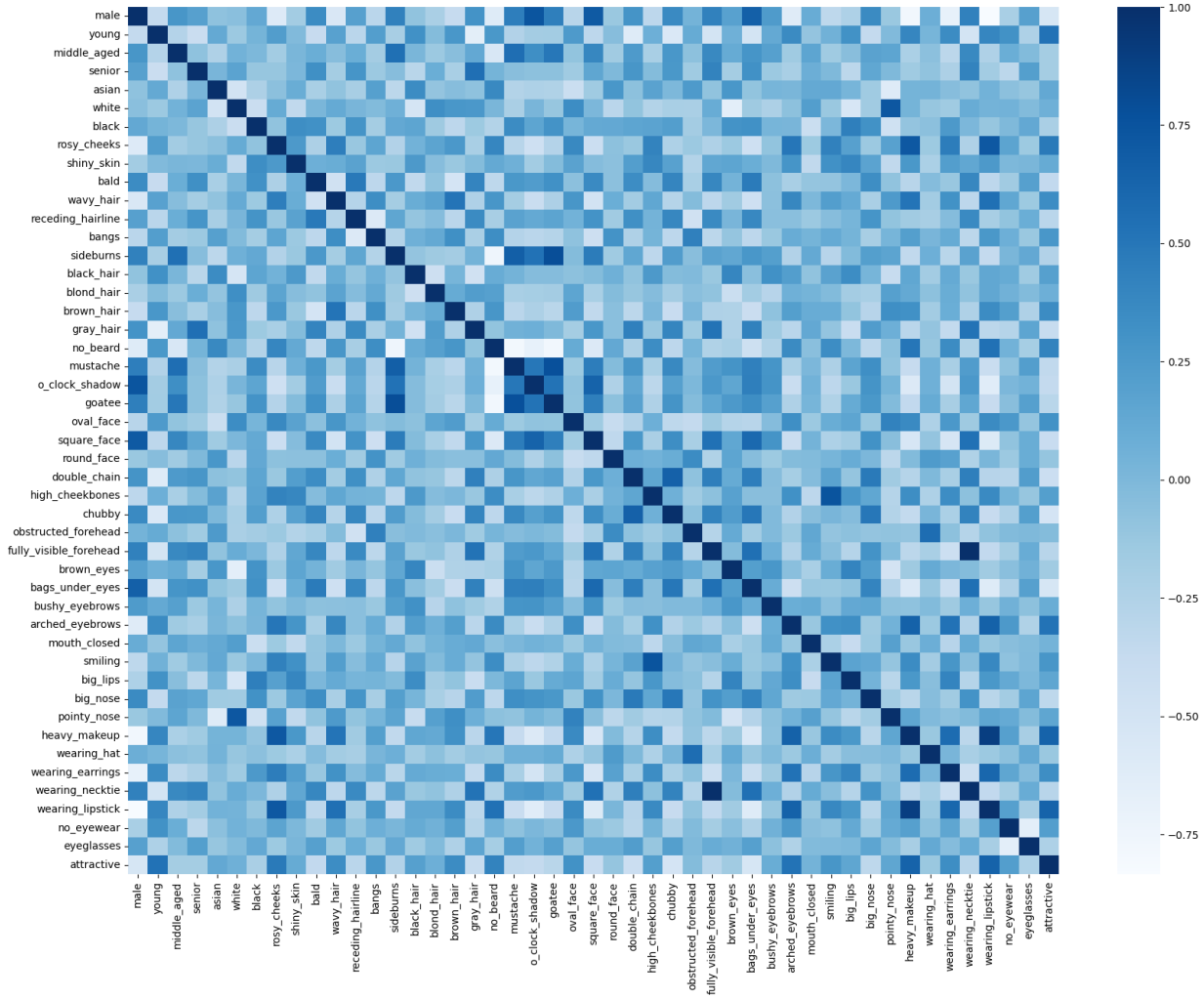


Figure 3.2: Correlation Matrix for the 47 attributes extracted with MAAD in BUPT-Balancedface.

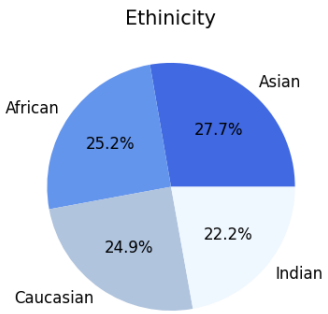


Figure 3.3: Ethnicity distribution in BUPT-Balancedface dataset.

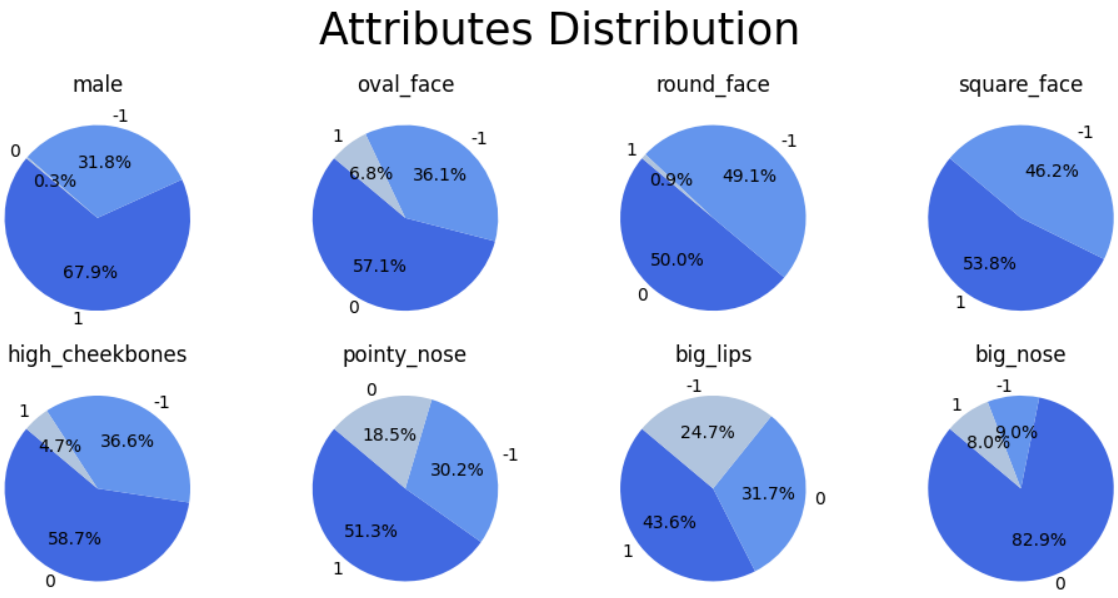


Figure 3.4: Distribution for the selected attributes of the annotations.

for testing. This corresponds to 1001343 embeddings for training, 123781 for validation and 126300 for testing.

The 512D contained 2498 fewer embeddings and 3 identities since these were extracted in distinct periods, after a removal of images with alignment errors conducted by a member of the research group.

3.1.2 Pre-processing

The pre-processing steps were applied to the data that was already divided.

The ethnicity labels were converted from categorical to numeric. Hence, the labels were mapped from *African*, *Asian*, *Caucasian* and *Indian* to 0, 1, 2 and 3.

Since the main goal was the location of the face attributes in the embeddings, the undefined class should be discarded. Additionally, the distributions from this label created more imbalance in the dataset distribution.

Each gender label corresponding to individual images was replaced by the mode for its identity. After, the remaining undefined labels were removed.

For the remaining labels, as it is possible to observe in Figure 3.4, there is a high presence of the undefined label, except for the *square face*. Hence, applying the mode did not exhibit significant variation in the attribute distributions. In this case, rather than its removal as a pre-processing step, which would lead to a large data decrease, a different approach was used for training with these values.

These labels were also converted to avoid negative values. Consequently, the 1 was kept as the positive class but the -1, the negative class, was converted to 0.

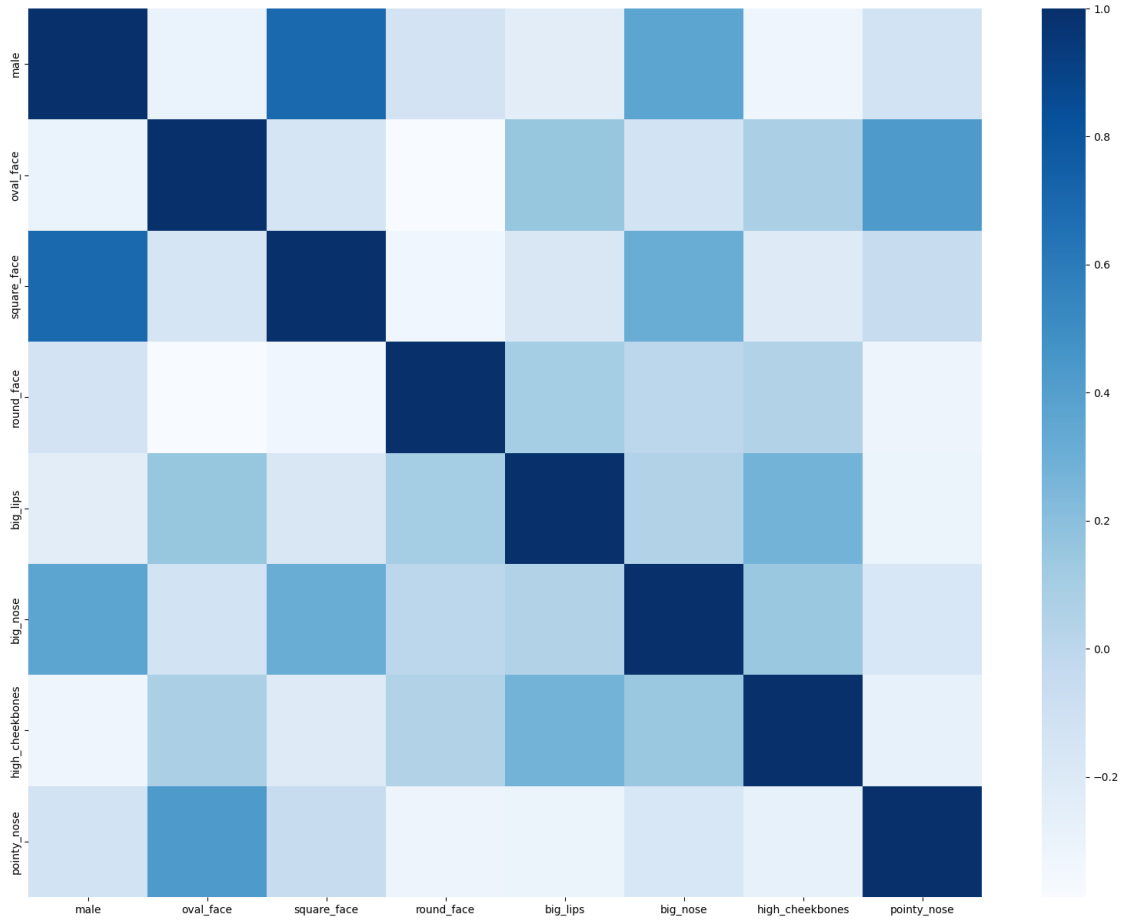


Figure 3.5: Correlation Matrix for the attributes used for classification.

3.2 Classifier Architecture Experiments

For the classifier, a multi-task approach was selected due to the advantage that could be gained from the joint feature learning. The implementation of the classifier was based on the one proposed for the MAC in [11], that is detailed in Chapter 2, in Subsection 2.3.3.

Different architectures were evaluated through ablation studies to determine which configuration yielded the most favourable results. Additionally, hyperparameter tuning was also performed. Both were held in a random search framework since the separation between training and testing identities needed to be guaranteed. In this case, using other frameworks such as grid-search would be more difficult to implement. This optimisation was implemented using the embeddings of size 128D and classifying only two attributes: gender and ethnicity.

3.2.1 Ablation Studies and Hyperparameter Tuning: Results and Discussion

Different architectures were implemented with one to two common hidden layers and zero to two task-specific hidden layers to assess which would lead to better results. All the hidden layers have ReLU activations, whereas output layers have softmax activation.

Figure 3.6 shows the different architectures implemented: A, B, C and D. The complexity of the model was increased until there were no gains in the model's performance.

Architecture A has two common layers (input and one hidden layer) and one task-specific output layer. Architecture B has one additional hidden specific layer, this corresponds to the architecture presented in [11]. Architecture C has three common layers, the input and two hidden layers and one branch layer that corresponds to the output layer. Finally, architecture D has one additional hidden layer compared to architecture B.

Cross-entropy was applied to each branch. Since it is a multi-task framework, the batch losses are computed with equal weight for the losses of each branch. The Adaptive Moment Estimation (Adam) algorithm was used as the optimiser. Early stopping was applied to avoid overfitting. The learning rate is defined as 1×10^{-4}, being chosen after hyperparameter tuning.

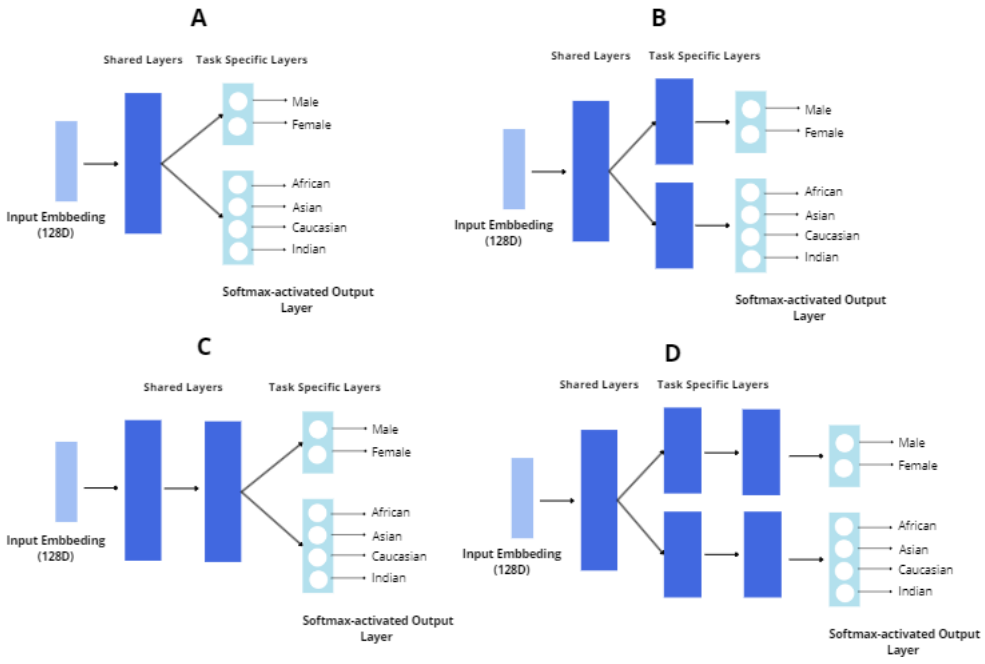


Figure 3.6: Ablation studies for the selection of the architecture for the classifier.

Table 3.1 shows the results for the different experiments for architecture A (only one common hidden layer). Different sizes for the hidden layer were implemented in a range from 32-128 nodes without dropout. Then, dropout was introduced and after, BatchNorm. With the introduction of dropout and BatchNorm, the performance decreased slightly. Nevertheless, with the increase in the number of nodes, the results also increased. Hence, BatchNorm, dropout and a higher number

of nodes were used for the following experiments. It is important to highlight that augmenting the number of nodes in the layers had a minimal impact on the training times.

Architecture	Loss	F1-Score Gender	F1-Score Ethnicity
32	0.1644	0.9716	0.9108
64	0.1609	0.9726	0.9124
128	0.1599	0.9727	0.9137
64, Dropout	0.1777	0.9712	0.9042
64, Dropout and BatchNorm	0.1863	0.9698	0.8986
128, Dropout and BatchNorm	0.1728	0.9719	0.9041
256, Dropout and BatchNorm	0.1610	0.9731	0.9138
512, Dropout and BatchNorm	0.1611	0.9737	0.9152
1024, Dropout and BatchNorm	**0.1572**	**0.9736**	**0.9170**

Table 3.1: Architecture A Experiments.

The batch size was selected by testing the performance of the best model of Architecture A (see Table 3.1). Three different sizes were tested: 16, 32 and 64. A batch size of 32 and 50 epochs showed a better performance. Therefore, these parameters were selected for further experiences. Table 3.2 shows the experiments for the different batch sizes.

Architecture	Loss	F1-Score Gender	F1-Score Ethnicity
Batch size: 16, 25 epoch	0.1642	0.9736	0.9129
Batch Size: 32, 50 epoch	**0.1572**	**0.9736**	**0.9170**
Batch Size: 64, 100 epoch	0.1584	0.9739	0.9161

Table 3.2: Batch Size Experiments.

Then, the impact of adding a hidden task-specific layer (Architecture B) was assessed. Table 3.3 shows the results for architectures with different node numbers. The architecture with a hidden layer of size 1024 and the task-specific layer of 512 was the one that presented the best results.

Architecture	Loss	F1-Score Gender	F1-Score Ethnicity
32, head 64	0.2281	0.9654	0.8875
64, head 128	0.1873	0.9711	0.8994
128, head 512	0.1664	0.9724	0.9093
512, head 256	0.1552	0.9741	0.9182
512, head 512	0.1529	0.9742	0.9173
512, head 1024	0.1529	0.9741	0.9180
1024, head 512	**0.1526**	**0.9747**	**0.9200**
1024, head 1024	0.1572	0.9747	0.9188

Table 3.3: Architecture B Experiments

Experiments with two common hidden layers (Architecture C) were also performed to check if extracting more common features from the embedding would lead to a better performance. Table 3.4 shows this experiment. However, the performance of this architecture was poorer than the ones obtained with Architecture B. Hence, this was discarded.

Architecture	Loss	F1-Score Gender	F1-Score Ethnicity
512 512	0.1540	0.9735	0.9179
1024 1024	**0.1536**	**0.9737**	**0.9190**

Table 3.4: Architecture C Experiments.

Finally, architecture D was tested, by adding a task-specific layer to Architecture B. The different experiments are shown in Table 3.5. The architecture with one common hidden layer of size 1024 and two task-specific layers of size 512 demonstrated the best performance. This was the architecture that had the best overall performance and the one selected to continue with further experiences.

Architecture	Loss	F1-Score Gender	F1-Score Ethnicity
1024, 512 512	**0.1483**	**0.9745**	**0.9210**
1024, 512 1024	0.1503	0.9745	0.9201
1024, 1024 1024	0.1528	0.9743	0.9181

Table 3.5: Architecture D Experiments.

3.2.2 Conclusions

These initial experiments showed interesting results when predicting gender and ethnicity. Nevertheless, the results for the gender were higher than the ones for ethnicity. This can be related to the gender task being binary, while the ethnicity is multi-class and therefore, harder to perform. Additionally, with simple architectures it was possible to predict both gender and ethnicity with an high accuracy.

With the ablation studies performed, the architecture selected to implement for the classification of face attributes was Architecture D, constituted by two common layers: the input and a hidden layer, and three task-specific layers: two hidden and the output layer. This architecture increased the depth of the work proposed in [11] by one layer and introduced one initial layer with a higher number of nodes.

With the increase in depth, it was possible to decrease the loss and increase the F1-score for the gender and ethnicity attributes. Nevertheless, the impact on the performance was not particularly significant possibly related to the high volume of data and the task's simplicity. Thus, the more complex architecture was chosen since it held better results and the training time was similar.

3.3 Multi-task Classifiers Implementation

Architecture D, with a common hidden layer of size 1024, and two hidden layers of size 512, was used to implement the four face attribute classifiers.

The models were trained with a batch size of 32 for 50 epochs and with the same parameters described in Subsection 3.2.1.

3.3.1 MTL1

The proposed architecture was trained with 128 and 512-sized embeddings for gender and ethnicity prediction, corresponding to models MTL1A and MTL1B. It should be noted that the results for the MTL1 model are different from the ones presented in the previous Section, even though it corresponds to the same model since the procedure was reproduced to introduce *seeds* that allow future reproducibility. Nevertheless, the difference in the performance is not significant.

3.3.1.1 Results and Discussion

Table 3.6 shows the F1-Scores for the MTL1A and MTL1B models. For both models, the performance for gender is higher than for ethnicity. The performance of MTL1B is higher than the one for MTL1A. This difference is larger for Ethnicity.

Architecture	Embedding Size	Loss	F1-Score Gender	F1-Score Ethnicity
MTL1A	128	0.1484	0.9742	0.9198
MTL1B	**512**	**0.1106**	**0.9869**	**0.9563**

Table 3.6: Comparison between MTL1A and MTLB results.

Figure 3.7 and Figure 3.8 represent the histogram of the accuracy for gender, ethnicity and average performance per identity for MTL1A and MTL1B, respectively. Such graphics give information on the performance per identity for testing identities. If the performance is low for specific identities, this can be related to poor examples, such as mislabelled or noisy inputs. On the other hand, if the model shows low accuracy for several identities, this can be related to issues with the model's implementation.

For MTL1A, the majority of the identities have performance above 88% for the average attribute, gender, and ethnicity accuracies per identity. For gender, identities that have a performance below 75%. Conversely, for ethnicity, as it was observed with F1-Score the performance is worse than it is for gender. It shows more identities with performance below 75%. Moreover, this attribute has some identities with performance lower than 50%.

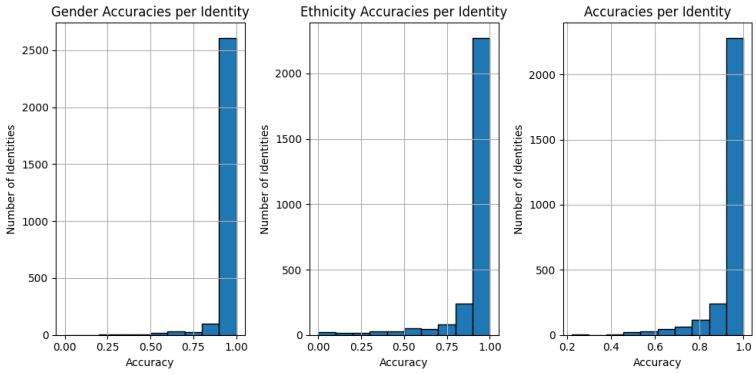


Figure 3.7: Histogram of the accuracies per identity for MTL1A.

For MTL1B, there is an increase in the performance per identity. This can be partially related to the removal of the 3 identities that had errors with the alignment of the images, but also to the higher number of features that can represent better the complex image data. Additionally, besides the different embedding sizes, different models were used for the extraction of the embeddings. The difference is more significant for ethnicity, leading to very few identities with an accuracy of less than 75%.

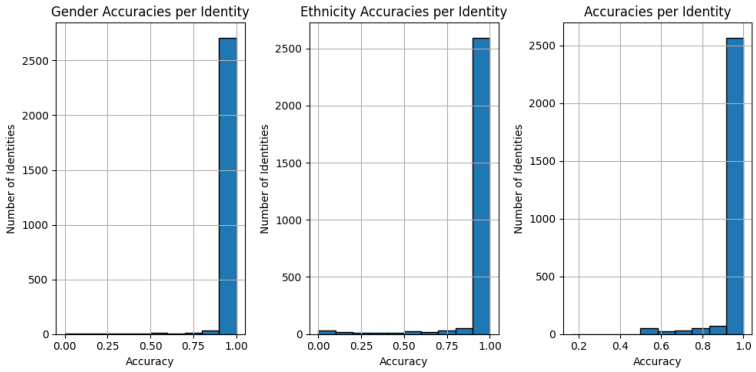


Figure 3.8: Histogram of the accuracies per identity for MTL1B.

The confusion matrices for the gender and ethnicity attribute predictions were computed. Analysing the confusion matrices for gender for MTL1A (see Figure 3.9) and MTL1B (see Figure 3.10), it is possible to observe that a good performance is achieved when classifying gender. However, the performance for class 0 (i.e., negative class, female) is worse. Additionally, it is possible to observe an increase in the correct predictions for both classes for MTL1B confusion matrix when compared with MTL1A and that the number of wrong predictions also decreases substantially. Particularly, the number of TN had an increase of around 1000 values with the corresponding decrease in FP. On the other hand, the number of TP increased by approximately 400 values with the FN decreasing a similar amount. This means that besides the increase in the general performance in MTL1B, it was also possible to increase the reliability in the positive and negative predictions, and this increase was more significant for the negative class (i.e., female), which is the minority class.

For ethnicity, the confusion matrices for MTL1A and MTL1B are represented in Figure 3.11 and Figure 3.12. It is possible to verify that there is an increase in the performance of the predictions for the classes 0 (African), 2 (Caucasian) and 3 (Indian). The confusion matrices for both models show a good performance. For MTL1A African-labelled embeddings are wrongly predicted as Caucasian and vice-versa. However, there are more African being predicted Caucasian than the opposite. Additionally, the Indian class was also incorrectly predicted as Caucasian. For MTL1B, the same behaviour is verified for the mislabeling of Africans with Caucasians and the opposite. For this model there is not a higher number of values from the Indian class to be predicted as Caucasian. Class 1 (Asian) is the one that has fewer incorrect predictions for both models.

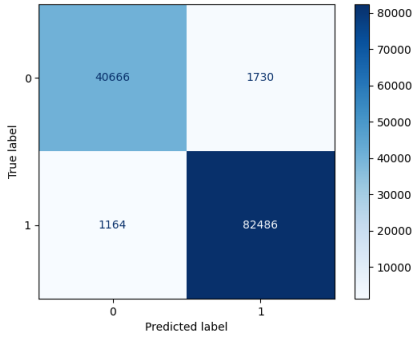


Figure 3.9: Confusion matrix for gender prediction with MTL1A.

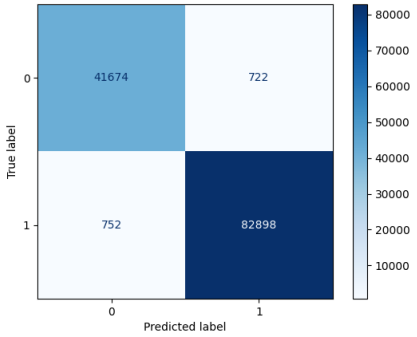


Figure 3.10: Confusion matrix for gender prediction with MTL1B.

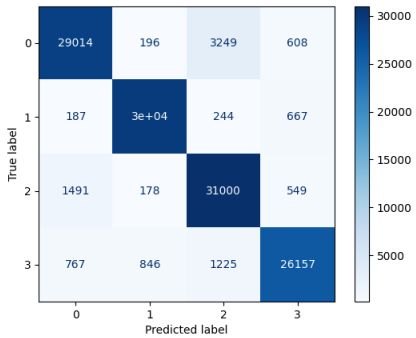


Figure 3.11: Confusion matrix for ethnicity prediction with MTLA.

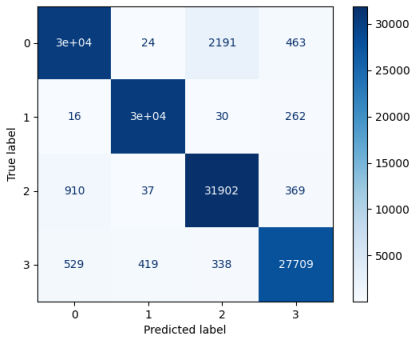


Figure 3.12: Confusion matrix for ethnicity prediction with MTLB.

Generally, both models present confusion matrices for the predictions of gender and ethnicity that show a desired performance.

3.3.1.2 Conclusions

The implementation of the selected architecture with embeddings of 128 and 512 dimensions showed good results when predicting gender and ethnicity. The performance increase for the model trained with embeddings extracted with *iResNet100*, MTL1B, and this growth was especially significant for ethnicity.

The analysis of the accuracy per identity showed that the majority of the identities has a great performance for both MTL1A and MTL1B. However, MTL1A has some identities with ethnicity accuracy lower than 50%. MTL1B shows a significant reduction in the accuracies per identity below 75% that lead to a better performance for the average accuracy per identity.

The gender confusion matrices computed for MTL1A and MTL1B demonstrated a higher performance for the male class than for the female which can be related to the class imbalance (higher number of male class labels). Moreover, MTL1B showed an increase in the prediction capacity when compared to MTL1A for both classes.

Finally, ethnicity confusion matrices revealed a higher performance from the Asian class (Class 1). They also revealed that there are some errors in predicting the Caucasian class as the African class and vice-versa. For MTLA there is also some wrong predictions for the Indian label, being predicted as Caucasian.

To summarise, the models show robust general and identity performance but still show some differences in the performance of the different classes for gender and ethnicity.

3.3.2 MTL2

Two different classifiers, MTL2A and MTL2B, were trained using 9 face attributes: gender, ethnicity, oval face, square face, round face, big nose, big lips, high cheekbones and pointy nose. The architecture for this model was the same as presented in Figure 3.6, Architecture D, but with 7 additional heads for the task-specific layers, corresponding to the introduced attributes.

As mentioned previously, there was a high presence of the undefined class for the majority of the remaining labels that were added later. The removal of these labels was not possible since this would lead to a reduced dataset for training, due to the scattered presence of the undefined labels across different embeddings annotations for different attributes. Thus, the strategy adopted was to remove class 2 (undefined) from the loss computation and define only two output nodes for this prediction. The idea was for the model to learn how to classify these attributes into positive (class 1) or negative (class 0) without removing the undefined class (class 2) from the dataset.

3.3.2.1 Results and Discussion

Table 3.7 displays the F1-scores obtained for the prediction of each attribute with MTL2A. The F1-Scores are computed using the predictions for the positive (1) and negative (0) classes. For

this model, the F1-Scores for the face, nose and lips shape attributes presented the highest results. Gender and Ethnicity had a minimal increase in performance compared to when their prediction was done without the rest of the attributes (MTL1A).

Gender	**Ethnicity**	**Oval Face**	**Round Face**	**Square Face**	**High Cheek-bones**	**Big Lips**	**Big Nose**	**Pointy Nose**
0.9747	0.9204	0.9999	1.0	0.9683	0.9983	0.9992	1.0	0.9990

Table 3.7: MTL2A performance per attribute.

Figure 3.13 shows the confusion matrices obtained for each predicted attribute. For gender the confusion matrix is very similar to the one predicted with MTL1A. There is a small increase in the performance of the class 0 (female). On the other hand, the performance for ethnicity also maintains the same conclusions of MTL1A, with a high number of African labels being classified as Caucasian labels and the opposite. Furthermore, the Indian labels have an elevated number of wrong predictions towards the Caucasian class.

The confusion matrices for the additional attributes include class 2 (undefined), but since the model was not trained for this class prediction, the predicted values for this class are all zero. Additionally, when the true label for the attributes is undefined, as the model was not designed to estimate that class, the predictions are divided between the other two classes.

Nevertheless, the positive (1) and negative (0) classes show great results. As for the square face attribute, there is a higher number of false positives and false negatives than for other attributes. This was the only additional attribute that was balanced for the positive and negative classes and did not include the undefined class.

Table 3.8 presents the results for MTL2B. The results for gender and ethnicity are similar to the ones obtained with MTL1B. However, the performance of the remaining attributes decreased with the training with these face templates. Figure 3.14 shows the confusion matrices for the attributes predicted with MTL2B.

For gender and ethnicity, the confusion matrices are closely aligned with the ones obtained from the results of MTL1B. The number of wrong predictions decreases when compared to the MTL2A. For ethnicity, the mislabelling of the Indian class as Caucasian is considerably lower when compared to MTL2A, as was observed with MTL1B and MTL1A.

For the other attributes, the confusion matrices reduced the performance when compared to MTL2A, with more incorrect predictions and fewer correct ones.

Gender	**Ethnicity**	**Oval Face**	**Round Face**	**Square Face**	**High Cheek-bones**	**Big Lips**	**Big Nose**	**Pointy Nose**
0.9872	0.9577	0.9709	0.9575	0.9016	0.9156	0.9283	0.9939	0.9266

Table 3.8: MTL2B performance per attribute.



Figure 3.13: Confusion matrix for multiple attribute prediction with MTL2A.

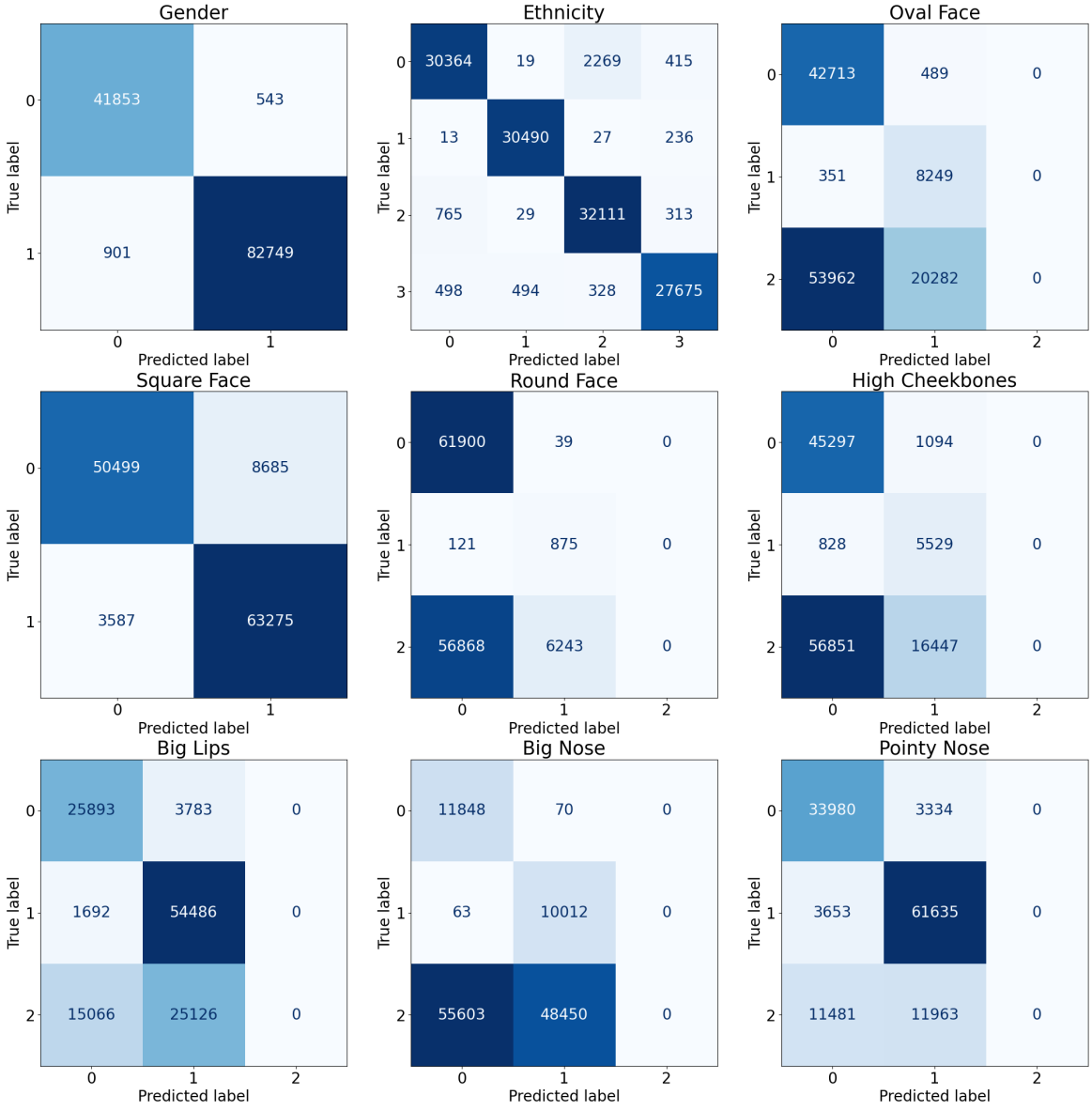


Figure 3.14: Confusion matrix for multiple attribute prediction with MTL2B.

3.3.2.2 Conclusions

The performance obtained for the predictions of gender and ethnicity with MTL2A and MTL2B models was slightly higher than the one obtained with MTL1A and MTL1B, respectively. This could mean that the joint estimation of these demographic attributes with the remaining attributes has a minimal impact. Additionally, the increase in the complexity of the model with the addition of more heads for predicting additional attributes could also make the new model more difficult to converge.

For MTL2A, the results for the additional face attributes were all above 96%. This could mean that the introduced soft-biometrics could benefit from the joint estimation of the attributes. Despite the class imbalance, the model showed a good performance for the positive and negative classes.

For MTL2B, the face geometry, mouth and lip shape performances were good but moderately smaller than the one obtained with MTL2A.

Chapter 4

Explaining Soft-Biometrics Predictions using SHAP Values

The SHAP values were implemented for the different classifiers to study the location of the soft biometrics in the embeddings, using the "shap" Python library[1] [65]. The "shap.GradientExplainer" was used to approximate the SHAP values for the impact of the features on the model's predictions. This method uses expected gradients (extension of integrated gradients) to approximate the SHAP values. With these SHAP values, the SHAP summary plots for each task of each model were plotted.

4.1 MTL1A

Figure 4.1 displays the SHAP summary plot for the gender classes predicted with MTL1A. It is possible to observe that the features with a higher importance for the model prediction are the same for both classes. The two summary plots are approximately symmetric. When a feature with a high value leads to a negative impact on the model's performance of a gender class, conversely, a high value of that feature will have a positive impact on the opposite class prediction.

Figure 4.2 presents the SHAP summary plot for the ethnicity prediction. Although the importance of the features is different for different ethnicity classes, there are common important features either between some of the classes or for all of the classes. Moreover, each class includes features with high impact that are only present for that class.

Features 33, 58, 59, 85 and 106 are features with high importance common for all the ethnicity classes. These features present different SHAP values for the different ethnicity classes. A negative and positive impact can be associated with the same feature level (high or low) for the same feature across the different classes. For instance, feature 33 has a negative impact on the model output for a high feature value and a positive one when this feature value is low, for all the ethnicity classes. On the other hand, the same level (high or low) of a certain important feature can have the opposite impact on different ethnicity classes. An example is that a negative impact by feature

[1] https://github.com/shap/shap/tree/master

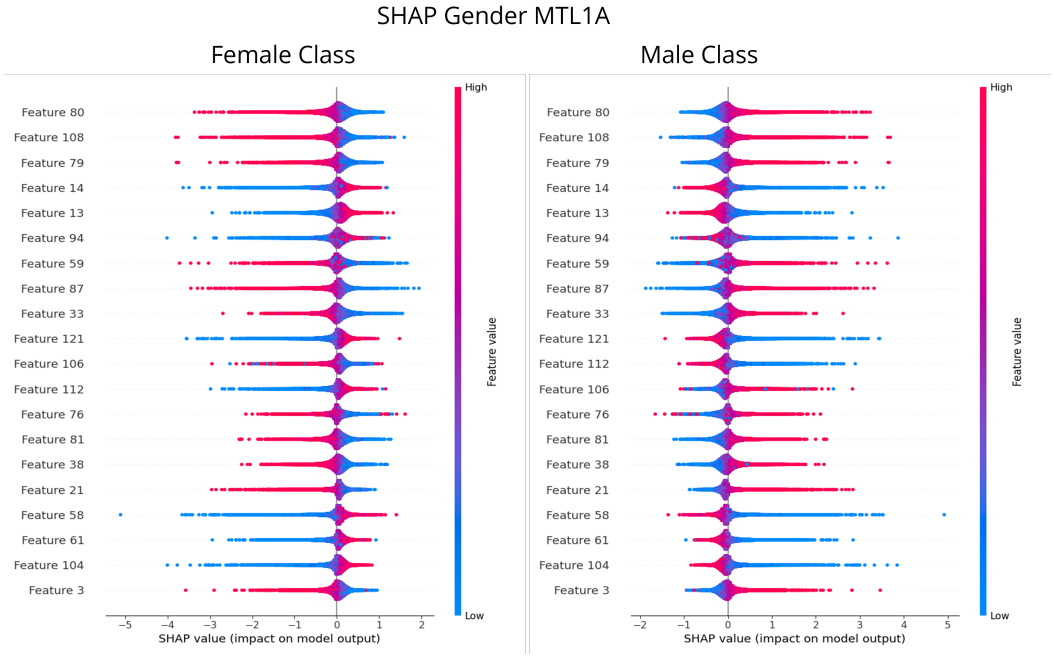


Figure 4.1: SHAP Summary plot for gender prediction of MTL1A.

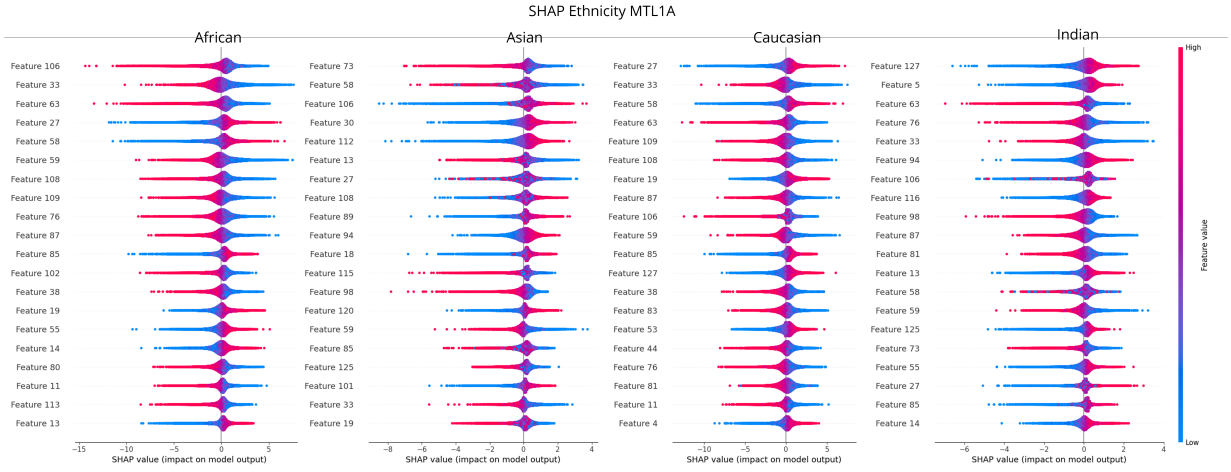


Figure 4.2: SHAP Summary plot for ethnicity prediction of MTL1A.

106 can be associated with either high or low feature values for different classes. Additionally, it is also possible to verify that the majority of these features, excluding feature 85, are included in the plot for the gender (see Figure 4.1).

Furthermore, African and Caucasian classes have features with a higher negative impact on the model output. These classes have 14 features of the 20 that correspond to the most important in common and with similar SHAP values and corresponding levels for the feature value.

For the Caucasian and the Indian classes, there are 11 common important features. For some features, the same feature level has a similar impact on the output of the model for the two classes. For other features, the same feature level can have different impacts on the predictions of the two different classes.

4.2 MTL1B

Similarly to MTL1A, the summary plots for each gender class are symmetric (see Figure 4.3). The input of this network has a higher dimension (512) than the input from MTL1A (128). The features with higher importance have locations scattered in the face template.

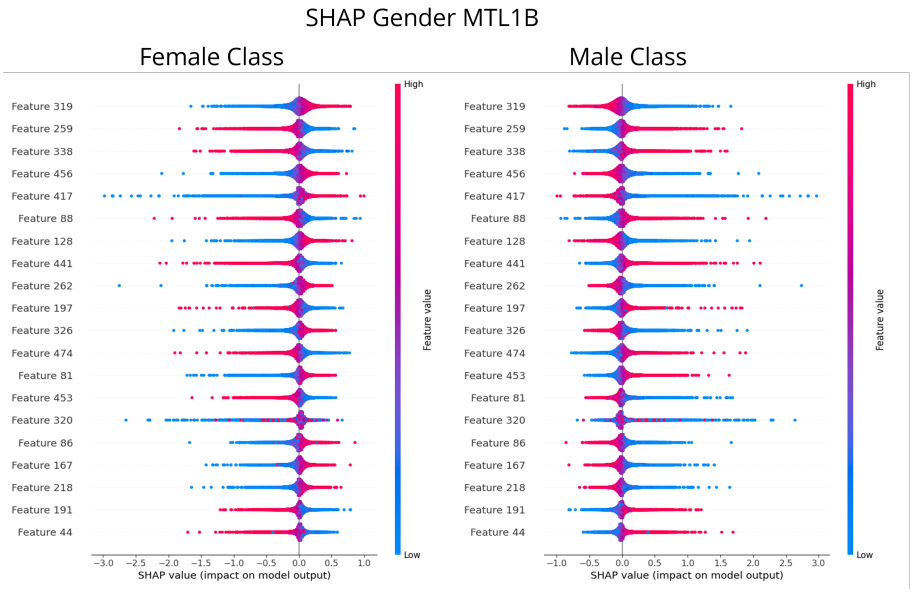


Figure 4.3: SHAP Summary plot for gender prediction of MTL1B.

For ethnicity, Figure 4.4, there are some common important features between the different groups of classes. Contrary to the analysis for the MTL1A ethnicity plot, there are no common features between all the classes.

Feature 116, for instance, has a high importance for the African, Asian and Caucasian classes. However, high feature values demonstrate a negative impact in the model prediction of the African class whereas, for the Asian class such range for the feature presents a positive impact on the output. The impact of this feature for Caucasian class is similar to the one for the African class.

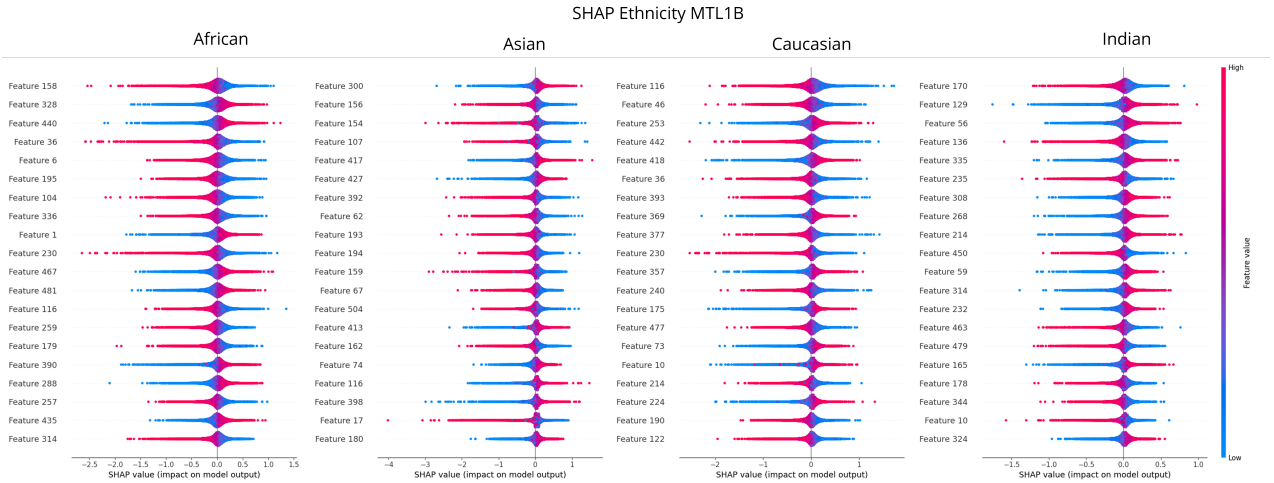


Figure 4.4: SHAP Summary plot for ethnicity prediction of MTL1B.

African and Caucasian plots also show that the features 36 and 230 have a high importance for both classes and a similar impact on the model output.

Features 10 and 214 are both present in the plots for Caucasian and Indian class. However, high values for the features demonstrate a negative impact on the prediction of the Caucasian class, whereas low feature values represent a negative contribution for the Indian class output. Feature 214 also impacts the classes differently: low values have a negative impact for Caucasian and high values have a negative impact on the prediction of Indian class.

Nevertheless, between the different classes, there are relevant features that are located nearby. For example, African class feature 158 is included in the listed features with the most global impact and the Asian class has features for location 154, 156 and 159.

4.3 MTL2A

Figure 4.5 represents the most important features and their respective impact in the prediction of the gender class with MTL2A. Comparing this Figure with the MTL1A gender prediction (see Figure 4.1) it is possible to observe that both share a great part of the features (Feature 3, 13, 14, 21, 33, 58, 59, 80,81, 106, 108) and that the impact of each value of the feature for the final output is similar.

For ethnicity (see Figure 4.6), there are also resemblances with the predictions of MTL1A. Most classes show the same important features as when ethnicity was only predicted jointly with gender, but some have some different features that can be related to learning different patterns by the addition of more features to predicted altogether. A significant proportion of the features has a similar impact on the model output when compared to ethnicity prediction on MTL1A. However, for the Asian class, both high and low features contribute to a certain feature having a positive or negative impact, something less noticeable for the MTL1A ethnicity prediction.

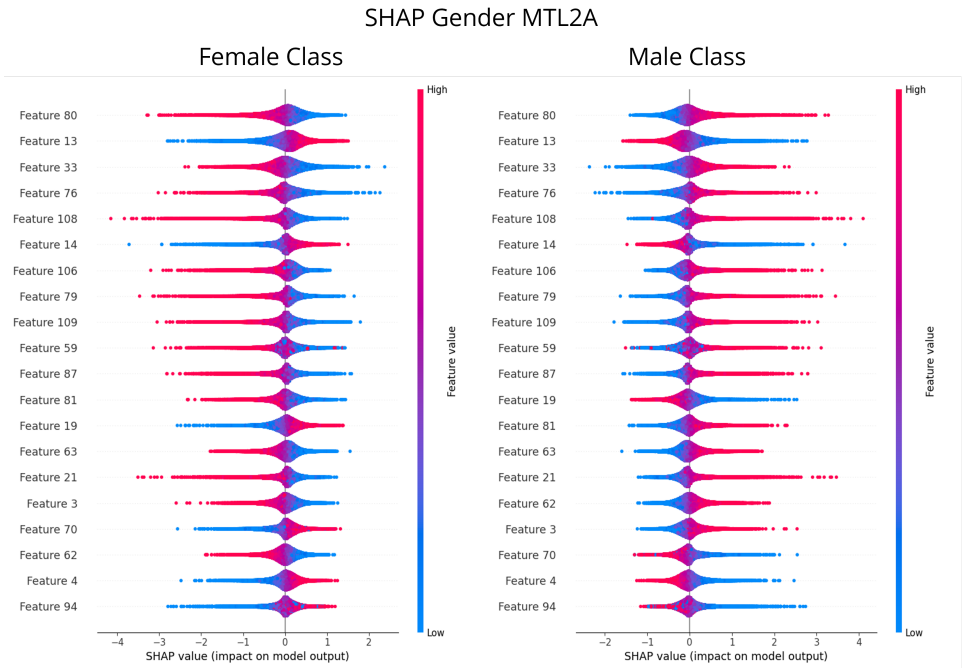


Figure 4.5: SHAP Summary plot for gender prediction of MTL2A.

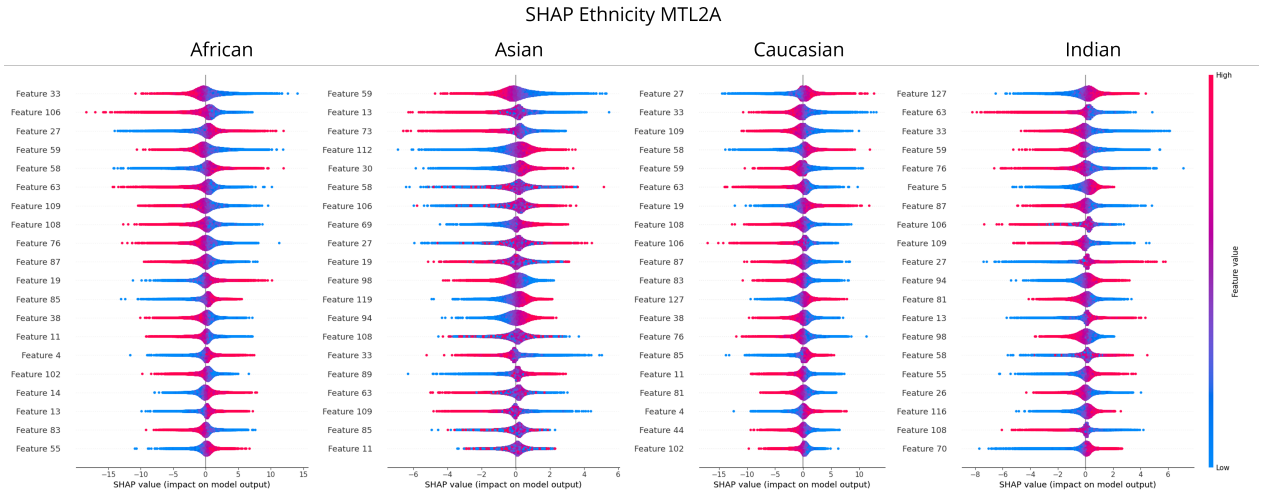


Figure 4.6: SHAP Summary plot for ethnicity prediction of MTL2A.

For the remaining attributes prediction, since this is also a binary task, the plots between the two different classes are also symmetric. Figure 4.7, 4.8, 4.9, 4.10 display the summary plots for the face shape, high cheekbones, nose shape and lip shape attributes, respectively. These share some common important features with both gender and some ethnicity classes. For instance, feature 13 has a high importance for the African, Asian and Indian class and for the gender prediction with MTL2A, and is present in the most important features for the prediction of high cheekbones (see Figure 4.8).

For the different face shape prediction tasks (see Figure 4.7) there are some common important features used for the predictions.

For the predictions related to the nose shape (see Figure 4.9), the prediction of a big nose and pointy nose share some important features, such as 20, 40, 42, and 53, as expected. Nevertheless, these have different magnitudes for the impact and some have different feature values for the same type of impact (negative or positive).

4.4 MTL2B

The SHAP summary plot for the prediction of gender with the MTL2B (see Figure 4.11) and MTL1B have similarities. Features 44, 88, 167, 197, 259, 338, and 417 are common for both. For MTL1B and MTL2B, the same features show a similar relation between a high or low feature value and their impact on the prediction of the model output. The predictions of class 0 and 1 are opposite, as verified in the other gender plots.

For ethnicity, some locations of the input embedding have similar importance for the same MTL1B and MTL2B classes, there is less equivalence when compared with the gender attribute. Additionally, as verified with MTLT1B, different classes have the same features listed as the most important. As an example, feature 36 has a negative impact in predicting both African, Asian and Caucasian classes when this feature value is high, but different values for the magnitude of the impact.

For the additional face attributes, contrary to the other binary tasks, the feature importance order differs between the negative and positive classes. Nevertheless, the differences are minimal. Figure 4.13, 4.14, 4.15 and 4.16 represent the summary plots for face shape, high cheekbone, lip shape and nose shape attributes, respectively. Through the analysis of the different plots, it is possible to understand that some common features are used for different tasks predictions.

For the different face shape attributes (oval, square and round), there are some common important features between the different tasks, however, these have different impacts on the prediction of the different classes. For instance, features 401 and 403 are common between the prediction of square face and round face labels.

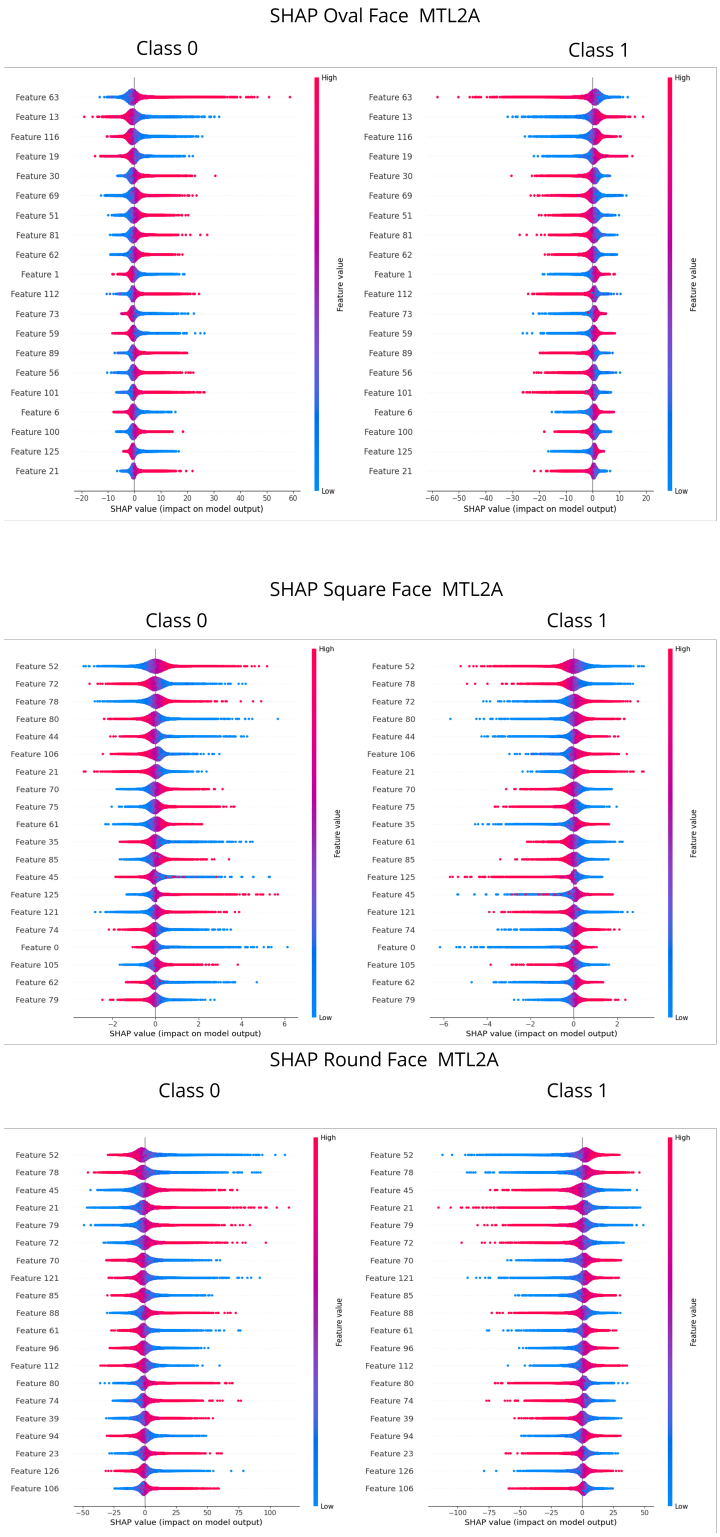


Figure 4.7: SHAP Summary plot for face shape prediction with MTL2A.

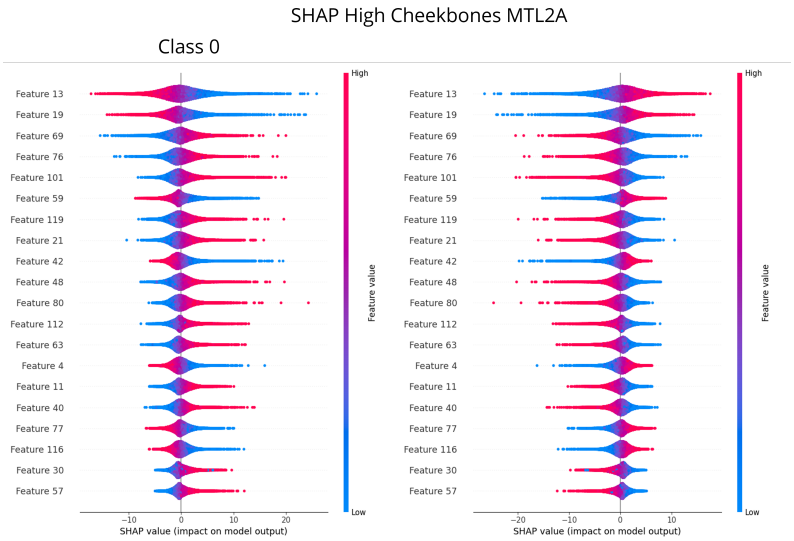


Figure 4.8: SHAP Summary plot for High Cheekbones prediction with MTL2A.

4.5 Conclusions

For the different models and attribute prediction tasks for each model, it was possible to obtain an explanation of the most important features, their impact on the output of the model and if this impact was related to a high or low value of the feature.

From the SHAP summary plots, it is possible to observe that for the models MTL1A, MTL1B, and MTL2A the binary predictions show an almost symmetric plot for the positive and negative classes, with the same order for the feature importance. Conversely, MTL2B shows some differences in the features that are important for a model prediction between two different classes of binary predictions.

On the other hand, for ethnicity, some of the classes have some common features and it is possible to compare the different influences these features can have on different classes.

Models MTL1A and MTL2A share common important features with similar impacts on the model output for gender and ethnicity prediction. MTL1B and MTL2B also share important features. This could be related to the presence of important features that are needed for performing the classification. The changed features can be related to adding more face attributes.

Different face attributes share the same important features. Between the same group of face attributes, for example, for the different groups in nose shape and face shape, there are also common important features, but this usually has different impacts on the model. Moreover, some of these tasks share common important features with gender and ethnicity tasks. These conclusions are verified for MTL2A and MTL2B.

The prediction of the different attributes is scattered across the embeddings. There is no attribute whose prediction relies mostly on a specific located set of features. On the other hand, features nearly located can have different output values.



Figure 4.9: SHAP Summary plot for nose attributes prediction with MTL2A.

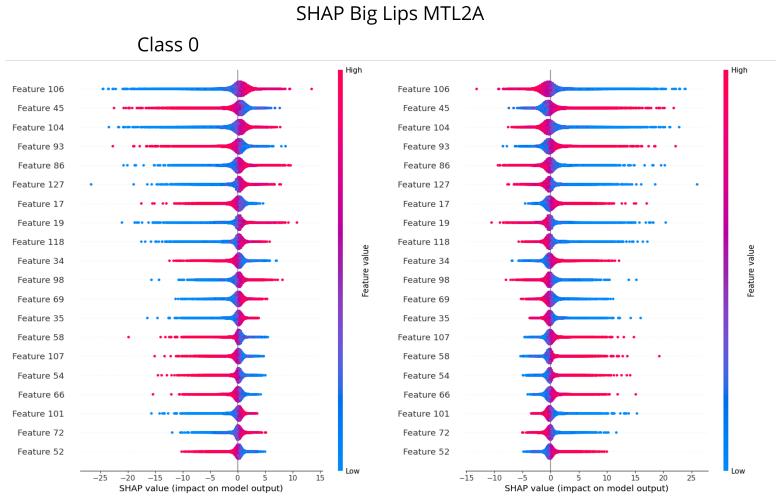


Figure 4.10: SHAP Summary plot for lip shape prediction with MTL2A.

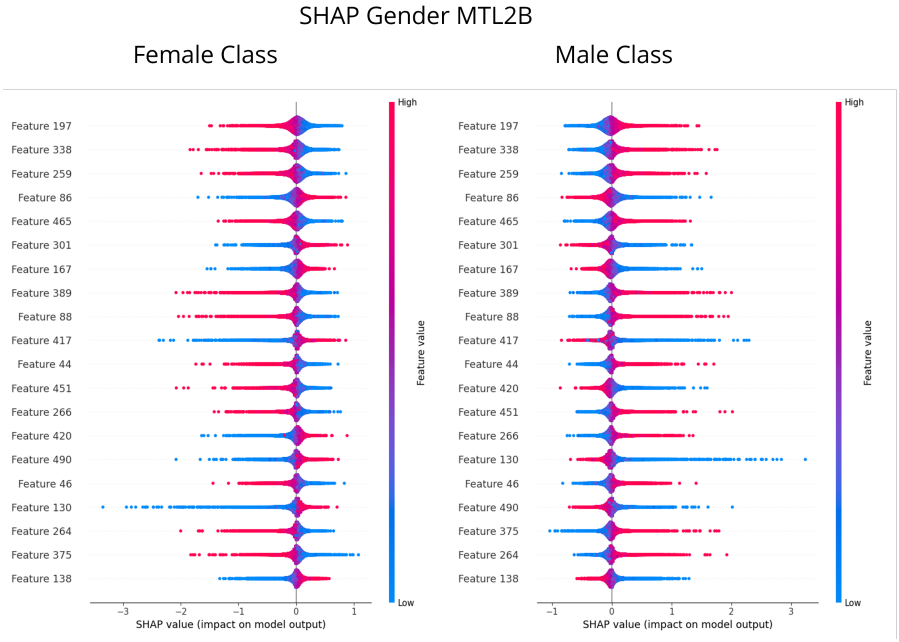


Figure 4.11: SHAP Summary plot for gender prediction of MTL2B.

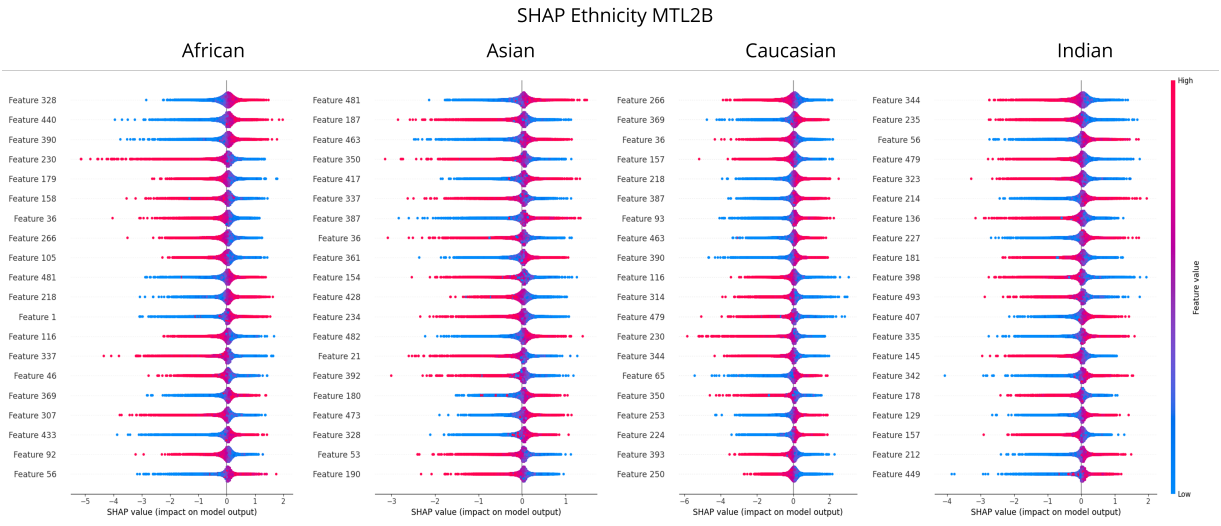


Figure 4.12: SHAP Summary plot for ethnicity prediction of MTL2B.

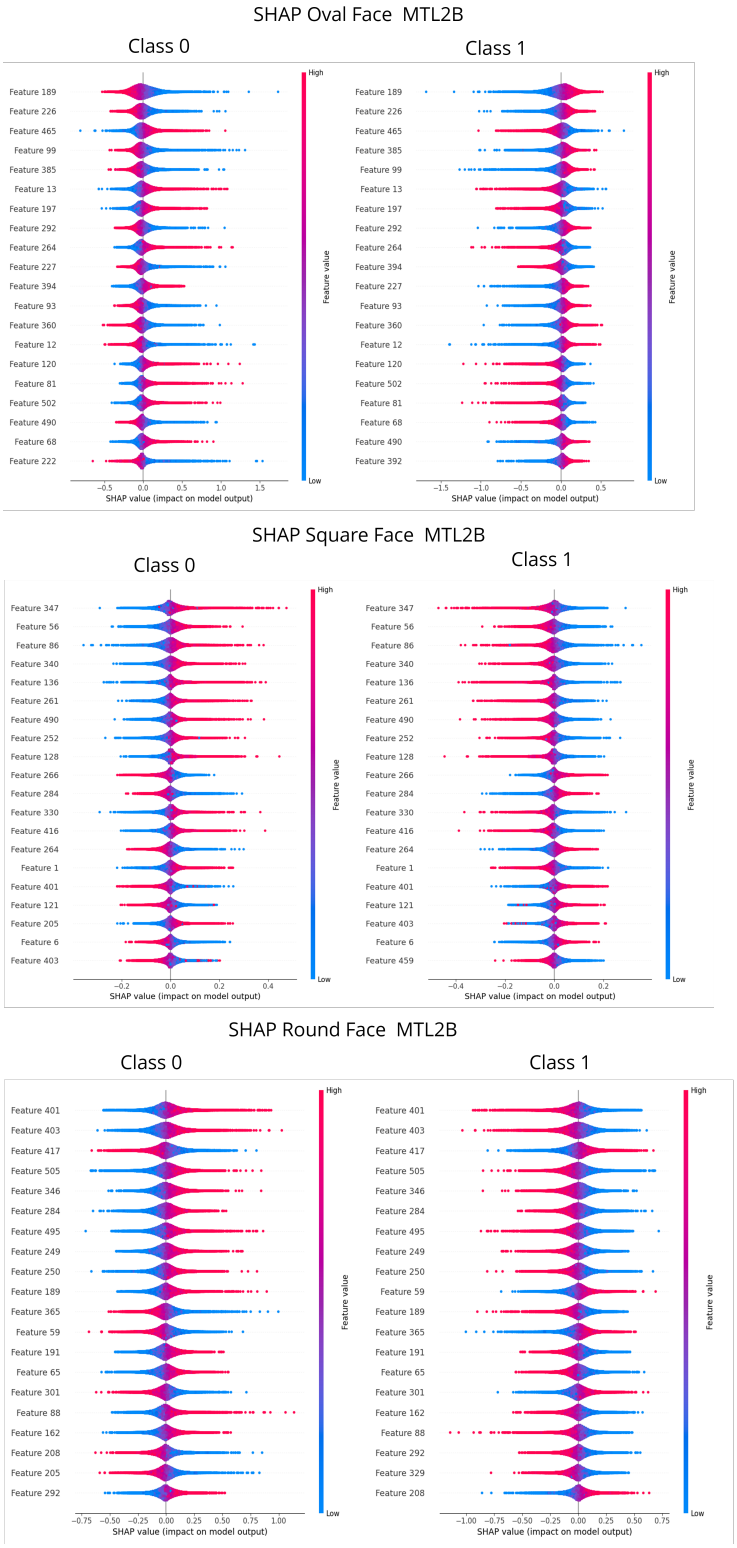


Figure 4.13: SHAP Summary plot for face shape prediction for MTL2B.

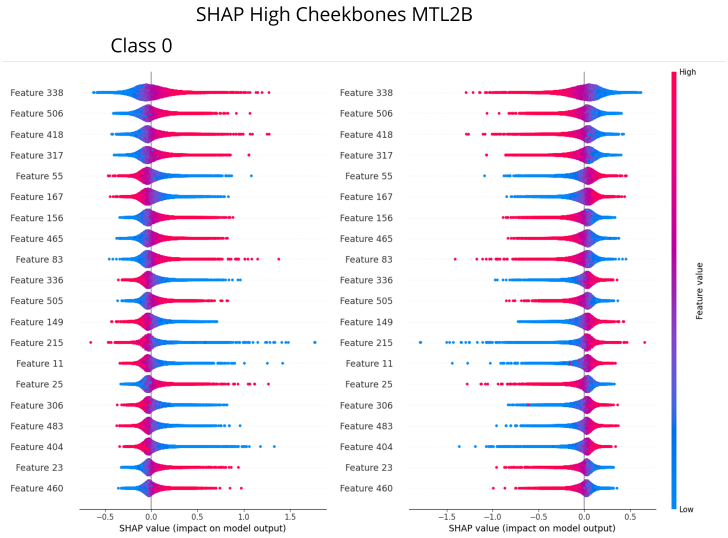


Figure 4.14: SHAP Summary plot for high cheekbone prediction for MTL2B.

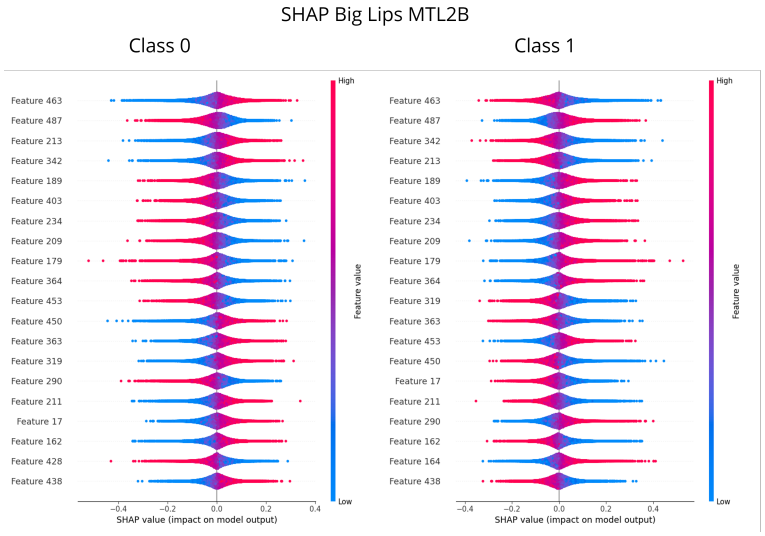


Figure 4.15: SHAP Summary plot for lip shape prediction for MTL2B.

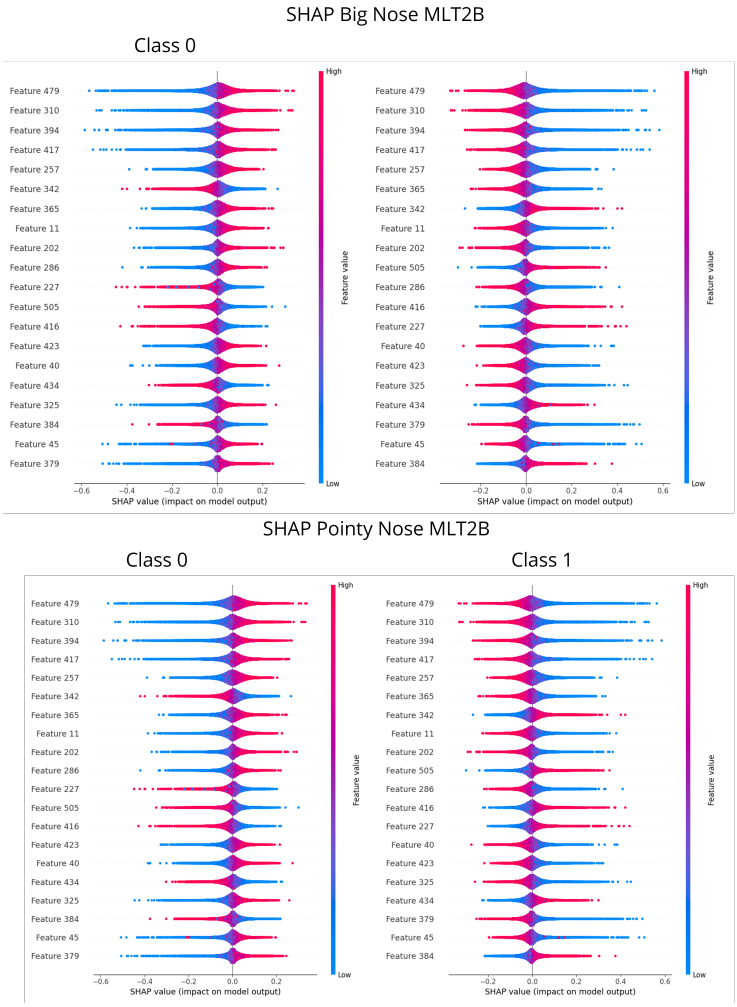


Figure 4.16: SHAP Summary plot for nose shape prediction for MTL2B.

Nevertheless, this approach only provides information about the location of the most important features and their value and if the feature magnitude is high or low for that impact. There is no interpretation of the reason for the model predicting that class.

Chapter 5

Conclusions and Future Work

5.1 Conclusions

With the developed work it was possible to train four different classifiers to jointly predict face attributes from face templates. These classifiers differ in the input size and origin and in the number of attributes that were predicted. These classifiers showed a good performance, meaning that such attributes can be successfully predicted from the face templates.

The architecture for the different models was obtained through ablation studies and hyperparameter tuning. There was not a significant difference between the performances, but the model with the best performance was selected since the higher complexity did not increase the training time. Additionally, it was possible to verify that a simple classifier with one layer could predict gender and ethnicity with a good performance.

The classifiers implemented to predict gender and ethnicity showed robust results. However, the results for the embeddings extracted with *iResNet100* (corresponding to B models, dimension of 512) presented a higher performance. Even though the dataset was balanced for ethnicity, it was not balanced for gender, so the results for the male gender were higher than for the female gender. For ethnicity, both classifiers demonstrated fewer errors for the Asian class and African labels were often mislabelled as Caucasian and vice-versa. Despite the good performance, the classifiers have different performances across different classes that can be either related to class imbalance, mislabelling or the quality of the dataset that was used.

The classifiers built to predict the nine face attributes also demonstrated good performance for the different attributes. The performance of the gender and ethnicity was similar to the one presented when predicting just the two face attributes. This could be due to the increase in the complexity of the model or related to the added face attributes not providing much additional information. For the classifier trained with the *iResNet100* embeddings (corresponding to B models, dimension of 512) the performance for some of the additional attributes was poorer.

The analysis of the SHAP summary plots for the predictions of the face attributes showed that the prediction of gender and ethnicity for MTL1A and MTL2A, and MTL1B and MTL2B, share common feature importance and these features have a similar impact on the output. Additionally,

different attributes share common features in the feature importance which can be related to the use of the multi-task framework. The important features for the prediction of the different classes for the different face attributes are scattered across the embedding and nearly located important features for a certain class can have different impacts on the model decision.

5.2 Future Work

With this study, it was possible to successfully implement face attribute classifiers that allowed the joint prediction of two and nine classifiers. Additionally, SHAP values were implemented for the attribute prediction of different classifiers. Some of the possible future research direction in the context of this work could be:

- Implementing single-task classifiers to compare the performance between STL and MTL. Despite the shared features between different tasks and improvement in the performance of some tasks when jointly predicting different attributes, studies that compared the prediction of the different attributes alone should be implemented to effectively assess the benefit of using an MTL approach.

- Increasing the number of predicted attributes. The ideal would be to predict the higher number of face attributes to allow the generation of complete explanations with the SHAP values. Additionally, adding more attributes could also be interesting for studying the effect on performance by jointly predicting several attributes. However, to perform this task, some of the labels of the used dataset such as "attractive" should be removed due to their subjectivity. Furthermore, new architectures could be studied for this higher number of face attributes to enhance the relation between attributes. For example, splitting the shared branches into categorical branches (e.g., face shape branch, nose shape branch) and then separating this into different branches for each attribute.

- Improve the dataset annotations. As mentioned in Subsection 3.1.2, the labels for the implemented dataset contained class imbalance and different attributes classified as undefined due to their origin from an automatic annotation pipeline. Hence, better pre-processing could be applied to improve these labels. On the other hand, a new automatic annotation pipeline could be used to improve these labels.

- Using SHAP values for implementing an explanation approach for FR. The SHAP values obtained for the proposed models could be used to leverage a new approach to FR that justified the difference between matching or not two faces with the differences in the SHAP for the soft-biometrics in the two different embeddings.

References

[1] United Nations, "Sustainable development goals." https://www.un.org/sustainabledevelopment, 2024. Accessed: 2024-09-10.

[2] I. Sarker, "Deep learning: A comprehensive overview on techniques, taxonomy, applications and research directions," *SN Computer Science*, vol. 2, 08 2021.

[3] S. Dong, P. Wang, and K. Abbas, "A survey on deep learning and its applications," *Computer Science Review*, vol. 40, p. 100379, 2021.

[4] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–44, 05 2015.

[5] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, no. 56, pp. 1929–1958, 2014.

[6] L. Alzubaidi, J. Zhang, A. J. Humaidi, A. Al-dujaili, Y. Duan, O. Al-Shamma, J. I. Santamaría, M. A. Fadhel, M. Al-Amidie, and L. Farhan, "Review of deep learning: concepts, cnn architectures, challenges, applications, future directions," *Journal of Big Data*, vol. 8, 2021.

[7] M. Grandini, E. Bagli, and G. Visani, "Metrics for multi-class classification: an overview," 2020.

[8] M. Wang and W. Deng, "Deep face recognition: A survey," *CoRR*, vol. abs/1804.06655, 2018.

[9] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, June 2015.

[10] F. Wang, H. Han, S. Shan, and X. Chen, "Deep multi-task learning for joint prediction of heterogeneous face attributes," in *2017 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017)*, pp. 173–179, 2017.

[11] P. Terhörst, D. Fährmann, J. N. Kolf, N. Damer, F. Kirchbuchner, and A. Kuijper, "Maad-face: A massively annotated attribute dataset for face images," 2021.

[12] P. C. Neto, T. Gonçalves, J. R. Pinto, W. Silva, A. F. Sequeira, A. Ross, and J. S. Cardoso, "Causality-inspired taxonomy for explainable artificial intelligence," 2024.

[13] S. M. Lundberg, G. G. Erion, and S. Lee, "Consistent individualized feature attribution for tree ensembles," *CoRR*, vol. abs/1802.03888, 2018.

[14] D. S. Trigueros, L. Meng, and M. Hartnett, "Face recognition: From traditional to deep learning methods," 2018.

[15] I. Adjabi, A. Ouahabi, A. Benzaoui, and A. Taleb-Ahmed, "Past, present, and future of face recognition: A review," *Electronics*, vol. 9, no. 8, 2020.

[16] Y. Kortli, M. Jridi, A. Al Falou, and M. Atri, "Face recognition systems: A survey," *Sensors*, vol. 20, no. 2, 2020.

[17] Y. Lu and T. Ebrahimi, "Explanation of face recognition via saliency maps," 2023.

[18] M. Huber, A. T. Luu, P. Terhörst, and N. Damer, "Efficient explainable face verification based on similarity score argument backpropagation," 2023.

[19] P. Terhörst, D. Fährmann, N. Damer, F. Kirchbuchner, and A. Kuijper, "Beyond identity: What information is stored in biometric face templates?," *CoRR*, vol. abs/2009.09918, 2020.

[20] A. B. Arrieta, N. Díaz-Rodríguez, J. D. Ser, A. Bennetot, S. Tabik, A. Barbado, S. García, S. Gil-López, D. Molina, R. Benjamins, R. Chatila, and F. Herrera, "Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai," 2019.

[21] A. Adadi and M. Berrada, "Peeking inside the black-box: A survey on explainable artificial intelligence (xai)," *IEEE Access*, vol. 6, pp. 52138–52160, 2018.

[22] D. V. Carvalho, E. M. Pereira, and J. S. Cardoso, "Machine learning interpretability: A survey on methods and metrics," *Electronics*, vol. 8, no. 8, 2019.

[23] T. A. John, V. N. Balasubramanian, and C. V. Jawahar, "Explaining deep face algorithms through visualization: A survey," 2023.

[24] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

[25] S. Pouyanfar, S. Sadiq, Y. Yan, H. Tian, Y. Tao, M. P. Reyes, M.-L. Shyu, S.-C. Chen, and S. S. Iyengar, "A survey on deep learning: Algorithms, techniques, and applications," *ACM Comput. Surv.*, vol. 51, sep 2018.

[26] G. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural computation*, vol. 18, pp. 1527–54, 08 2006.

[27] I. Sarker, "Machine learning: Algorithms, real-world applications and research directions," *SN Computer Science*, vol. 2, 03 2021.

[28] G. Guo and N. Zhang, "A survey on deep learning based face recognition," *Computer Vision and Image Understanding*, vol. 189, p. 102805, 08 2019.

[29] W. Rawat and Z. Wang, "Deep convolutional neural networks for image classification: A comprehensive review," *Neural Computation*, vol. 29, pp. 1–98, 06 2017.

[30] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'12, (Red Hook, NY, USA), p. 1097–1105, Curran Associates Inc., 2012.

[31] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, vol. abs/1409.1556, 2014.

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

[33] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.

[34] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7132–7141, 2018.

[35] A. K. Jain, P. Flynn, and A. A. Ross, *Handbook of Biometrics*. Springer Publishing Company, Incorporated, 1st ed., 2010.

[36] P. C. Neto, R. M. Mamede, C. Albuquerque, T. Gonçalves, and A. F. Sequeira, "Massively annotated datasets for assessment of synthetic and real data in face recognition," *arXiv preprint arXiv:2404.15234*, 2024.

[37] R. Ranjan, S. Sankaranarayanan, A. Bansal, N. Bodla, J.-C. Chen, V. M. Patel, C. D. Castillo, and R. Chellappa, "Deep learning for understanding faces: Machines may be just as good, or better, than humans," *IEEE Signal Processing Magazine*, vol. 35, no. 1, pp. 66–83, 2018.

[38] A. K. Jain and S. Z. Li, *Handbook of Face Recognition*. Berlin, Heidelberg: Springer-Verlag, 2005.

[39] Y. Sun, X. Wang, and X. Tang, "Hybrid deep learning for face verification," in *2013 IEEE International Conference on Computer Vision*, pp. 1489–1496, 2013.

[40] Y. Sun, X. Wang, and X. Tang, "Deep learning face representation from predicting 10,000 classes," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1891–1898, 2014.

[41] Y. Guo, L. Zhang, Y. Hu, X. He, and J. Gao, "Ms-celeb-1m: A dataset and benchmark for large-scale face recognition," *CoRR*, vol. abs/1607.08221, 2016.

[42] D. Yi, Z. Lei, S. Liao, and S. Z. Li, "Learning face representation from scratch," *CoRR*, vol. abs/1411.7923, 2014.

[43] O. Parkhi, A. Vedaldi, and A. Zisserman, "Deep face recognition," in *BMVC 2015-Proceedings of the British Machine Vision Conference 2015*, British Machine Vision Association, 2015.

[44] Q. Cao, L. Shen, W. Xie, O. M. Parkhi, and A. Zisserman, "Vggface2: A dataset for recognising faces across pose and age," *CoRR*, vol. abs/1710.08092, 2017.

[45] G. Huang, M. Mattar, T. Berg, and E. Learned-Miller, "Labeled faces in the wild: A database for studying face recognition in unconstrained environments," *Tech. rep.*, 10 2008.

[46] M. Wang, Y. Zhang, and W. Deng, "Meta balanced network for fair face recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, p. 1–1, 2021.

[47] A. Hasnat, J. Bohné, S. Gentric, and L. Chen, "Deepvisage: Making face recognition simple yet with powerful generalization skills," *CoRR*, vol. abs/1703.08388, 2017.

[48] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "Deepface: Closing the gap to human-level performance in face verification," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1701–1708, 2014.

[49] Y. Sun, X. Wang, and X. Tang, "Deep learning face representation by joint identification-verification," 2014.

[50] Y. Sun, X. Wang, and X. Tang, "Deeply learned face representations are sparse, selective, and robust," *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2892–2900, 2014.

[51] Y. Sun, D. Liang, X. Wang, and X. Tang, "Deepid3: Face recognition with very deep neural networks," *CoRR*, vol. abs/1502.00873, 2015.

[52] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," *CoRR*, vol. abs/1311.2901, 2013.

[53] W. Liu, Y. Wen, Z. Yu, M. Li, B. Raj, and L. Song, "Sphereface: Deep hypersphere embedding for face recognition," *CoRR*, vol. abs/1704.08063, 2017.

[54] J. Deng, J. Guo, and S. Zafeiriou, "Arcface: Additive angular margin loss for deep face recognition," *CoRR*, vol. abs/1801.07698, 2018.

[55] N. Kumar, A. C. Berg, P. N. Belhumeur, and S. K. Nayar, "Attribute and simile classifiers for face verification," in *2009 IEEE 12th International Conference on Computer Vision*, pp. 365–372, 2009.

[56] E. M. Hand and R. Chellappa, "Attributes for improved attributes: A multi-task network for attribute classification," *CoRR*, vol. abs/1604.07360, 2016.

[57] K. Nandakumar and A. K. Jain, *Soft Biometrics*, pp. 1235–1239. Boston, MA: Springer US, 2009.

[58] H. Han, A. K. Jain, F. Wang, S. Shan, and X. Chen, "Heterogeneous face attribute estimation: A deep multi-task learning approach," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 11, pp. 2597–2609, 2018.

[59] Z. Liu, P. Luo, X. Wang, and X. Tang, "Deep learning face attributes in the wild," in *2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 3730–3738, 2015.

[60] K. Ricanek and T. Tesafaye, "Morph: A longitudinal image database of normal adult age-progression," vol. 2006, pp. 341 – 345, 05 2006.

[61] M. Ehrlich, T. J. Shields, T. Almaev, and M. R. Amer, "Facial attributes classification using multi-task representation learning," in *2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 752–760, 2016.

[62] P. P. Angelov, E. A. Soares, R. Jiang, N. I. Arnold, and P. M. Atkinson, "Explainable artificial intelligence: an analytical review," *WIREs Data Mining and Knowledge Discovery*, vol. 11, no. 5, p. e1424, 2021.

[63] F. Doshi-Velez and B. Kim, "Towards a rigorous science of interpretable machine learning," 2017.

[64] J. Pearl and D. Mackenzie, *The Book of Why: The New Science of Cause and Effect*. USA: Basic Books, Inc., 1st ed., 2018.

[65] S. M. Lundberg and S. Lee, "A unified approach to interpreting model predictions," *CoRR*, vol. abs/1705.07874, 2017.

[66] V. Petsiuk, A. Das, and K. Saenko, "RISE: randomized input sampling for explanation of black-box models," *CoRR*, vol. abs/1806.07421, 2018.

[67] J. Zhang, Z. Lin, J. Brandt, X. Shen, and S. Sclaroff, "Top-down neural attention by excitation backprop," *CoRR*, vol. abs/1608.00507, 2016.

[68] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2921–2929, 2016.

[69] R. R. Selvaraju, A. Das, R. Vedantam, M. Cogswell, D. Parikh, and D. Batra, "Grad-cam: Why did you say that? visual explanations from deep networks via gradient-based localization," *CoRR*, vol. abs/1610.02391, 2016.

[70] M. Sundararajan, A. Taly, and Q. Yan, "Axiomatic attribution for deep networks," *CoRR*, vol. abs/1703.01365, 2017.

[71] J. R. Williford, B. B. May, and J. Byrne, "Explainable face recognition," *CoRR*, vol. abs/2008.00916, 2020.