BABEŞ-BOLYAI UNIVERSITY CLUJ-NAPOCA
FACULTY OF MATHEMATICS AND COMPUTER SCIENCE
SPECIALIZATION COMPUTER SCIENCE IN ENGLISH

DIPLOMA THESIS

MPAIFR: Missing Persons Age-Invariant Face Recognition

Supervisor
Lecturer, PhD. Borza Diana-Laura

Author
Rusu Raluca-Maria

2024

UNIVERSITATEA BABEŞ-BOLYAI CLUJ-NAPOCA
FACULTATEA DE MATEMATICĂ ŞI INFORMATICĂ
SPECIALIZAREA INFORMATICĂ ÎN LIMBA ENGLEZĂ

LUCRARE DE LICENŢĂ

MPAIFR: Identificarea Persoanelor Dispărute prin Recunoaștere Facială Invariabilă de Vârstă

Coordonator științific
Lect. Dr. Borza Diana-Laura

Absolvent
Rusu Raluca-Maria

2024

ABSTRACT

Interest in Age-Invariant Face Recognition research continues to increase due to the challenging problem of matching faces with significant age differences because of the variations in facial features across aging. Addressing this issue is vital as traditional Face Recognition systems often fail to recognize individuals accurately.

To reduce this disparity, this thesis proposes a method to eliminate age-reliant and gender-reliant elements from features containing identity, age, and gender information. Particularly, the face features are factorized into three uncorrelated elements: identity-reliant, age-reliant, and gender-reliant. The essential data required for effective facial recognition relies on the identity-reliant element. A Decorrelated Adversarial Learning (DAL) algorithm is employed in the implementation, together with a Batch Canonical Mapping Module (BCMM), to determine the highest correlation between features produced by the Backbone Network. The Feature Residual Factorization Module (FRFM) and the Backbone Network are trained to generate features that minimize this correlation at the same time. Under the supervision of corresponding identity, age, and gender preserving signals, this approach ensures that the identity-reliant, age-reliant, and gender-reliant attributes are considerably decorrelated and contain the proper information. The recognition accuracy achieved on the FG-NET dataset is 94.61%.

To demonstrate the practical applicability of the developed model, it was integrated into an API, providing a flexible and efficient interface for developers. This enables the incorporation of AIFR capabilities into various applications. Additionally, a web application was developed to assist in the search for missing persons. This application allows users to access profiles of missing individuals and submit inquiries with images of potential matches. Law enforcement admin users can review these inquiries, view calculated similarity scores from the model, and the label given based on the likelihood of a match, thereby enhancing the search and identification process.

The accuracy and reliability of the AIFR model and the features of the API and web application can enhance the effectiveness of search and identification efforts, potentially reuniting families and improving public safety, but also elevate the collaboration between civilians and law enforcement. The integration also demonstrates its practical utility and potential for widespread adoption.

Contents

Chapter 1

Introduction

1.1 Context and motivation

The demand for improved security measures, identity systems, and personal authentication technologies has led to significant breakthroughs in the field of Face Recognition (FR) in recent years. Despite these advancements, the ability to recognize faces across different ages remains a persistent challenge. When there are large age disparities between the reference and query photos, traditional FR algorithms frequently have trouble matching faces correctly. The limitation is particularly problematic for circumstances in which an individual's appearance may have changed significantly over time, such as the search for missing persons.

The goal of Age-Invariant Face Recognition (AIFR) is to create models that can reliably identify people as they age in order to overcome this problem. This research is driven by the critical need for reliable recognition systems in order to help law enforcement and other relevant parties find missing people. By improving the accuracy and robustness of recognition across ages, AIFR can play a pivotal role in reuniting families and ensuring public safety, but also in the collaboration between civilians and law enforcement.

1.2 Objectives

The primary objective of this thesis is to conduct a comprehensive investigation into the challenges inherent in Age-Invariant Face Recognition and to develop robust solutions to address these challenges. Specifically, this research aims to examine the factors that impede Face Recognition across different age groups, including the variations in facial features due to aging and the impact of these variations on the performance of existing FR systems.

This investigation will lead to the design and implementation of an AI model

tailored for AIFR. The model will incorporate innovative techniques to enhance its ability to accurately identify individuals despite significant age differences between the target and test images, by decomposing the features into three uncorrelated elements: identity-reliant, age-reliant, and gender-reliant, and performing either a minimizing or a maximaning action to the correlation between the elements.

Furthermore, the thesis seeks to integrate the most effective AIFR model into an REST API, demonstrating its practical applicability across various use cases. The API will facilitate seamless integration into different systems that require reliable AIFR capabilities.

In addition, a web application will be developed to assist in locating missing persons by leveraging the model. This application will enable users to access profiles of missing individuals, including photographs and some personal information, and to submit inquiries accompanied by images of potential matches. Admin users will have the capability to review these inquiries, view the calculated similarity scores, and the label given based on the likelihood of a match. This research aims to assess the effectiveness of the developed web application in real-world scenarios, exploring its potential utility for law enforcement agencies in enhancing their search and identification efforts for missing persons.

By addressing these objectives, this thesis aims to contribute significant advancements to the field of Face Recognition technology, particularly in the context of age-invariance, and to provide practical tools that can assist in the critical task of locating missing individuals.

Our main contributions include the Decorrelated Adversarial Learning (DAL) algorithm based on the linear feature factorization to regularize the learning of the three decomposed elements: age, gender and identity. Our goal is to obtain age and gender invariant and identity-preserving characteristics for AIFR. To our knowledge, this is the first attempt to include the Decorrelated Adversarial Feature Learning for three facial characteristics. Another major contribution is the public REST API that incorporates our AIFR model. This proves that the model can be utilized across various platforms and use cases, providing a flexible interface for developers to implement Age-Invariant Face Recognition capabilities within their applications. Lastly, the web application designed to aid law enforcement in the search for missing persons underscores the social relevance and utility of the research conducted.

1.3 Thesis structure

The thesis is organized into multiple sections, each of which focuses on a distinct component of the research conducted. An overview of related research in the field is detailed in the first chapter. This provides a thorough overview of the literature on

Face Recognition and Age-Invariant Face Recognition, highlighting recent research and paving the context for future improvements. The theoretical underpinnings that were necessary to carry out this research are covered in the subsequent chapter.

Following these sections, the thesis delves into the methodology and development of the proposed AIFR model and technology for finding missing persons. It covers a number of topics, such as the difficulties faced, the architectural framework selected, and the specifics of its implementation. This part exposes an extensive overview of the methods and technologies used, highlighting their importance within the study.

The evaluation and findings of the experiment are presented in the next section of the thesis. This section offers an understanding of the experimental setup and the choices and methods used throughout this research approach. The experiment findings are also detailed, providing insight into the functionality and efficacy of the developed methods.

A thorough conclusion is provided in the thesis's last section, which is founded on the research's observations and conclusions. The primary findings and conclusions from the investigation are outlined in the conclusion. It also looks at various possibilities for the research's future growth and extension.

This methodical approach guarantees that the thesis provides a comprehensive examination of the matter. By integrating theoretical foundations, outcomes of experiments, technical aspects, and prospective directions, it offers a thorough knowledge of the research conducted and paves the path for future advancements in this area.

Chapter 2

Literature Review and Related Work

This chapter surveys developments in Biometrics and Computer Vision, specifically Face Recognition and the subtask of Age-Invariant Face Recognition. Through analyzing existing literature, we present the most important concepts, experiments, and discoveries. We analyze the details and particularities of the most widely used databases and review the current State-of-the-Art models and methods for Age-Invariant Face Recognition. Furthermore, the issues and challenges of this field, particularly of Age-Invariant Face Recognition, are discussed.

2.1 Face Recognition (FR)

A key characteristic that people use to identify one another is their face. Face Recognition (FR) has been an extensively researched topic for many years in Biometrics and Computer Vision. It represents the process of positively identifying a face in a picture or video by comparing it to a database of faces that already exist and are considered correct. It starts with image processing techniques like detection and alignment, which involves spotting and cropping faces from other things in the picture and aligning the face to the center of the image. Next, a feature extractor is used to excerpt characteristics from the photos and the algorithm focuses on matching the discovered features, by comparison to the database of faces [GZ19]. Neural Networks (NN), particularly Deep Neural Networks (DNN), have recently demonstrated remarkable results with vast training datasets and computational resources. By synthesizing complex characteristics out of small amounts of pixels, Deep Learning (DL) uses different Neural Networks and vast amounts of data to depict the core characteristics. These advanced significantly Face Recognition.

As computer hardware and imaging technologies have advanced, Face Recognition has become increasingly frequent in our day-to-day lives. Passport identity airport verification, card creation verification, video surveillance [GZ19], biometric

authentication [MMS19], and medical diagnostics are just a few of the numerous applications that employ FR technology today.

As the development of Face Recognition grew quickly, so did the demands. Humans can recognize hundreds of faces via lifelong learning and even recognize faces they haven't seen in a few years. We possess this aptitude to such a reasonable degree that it is rarely impaired by the passage of time or different visual changes brought on by aging, facial expressions, or factors like hair color or wearing glasses.

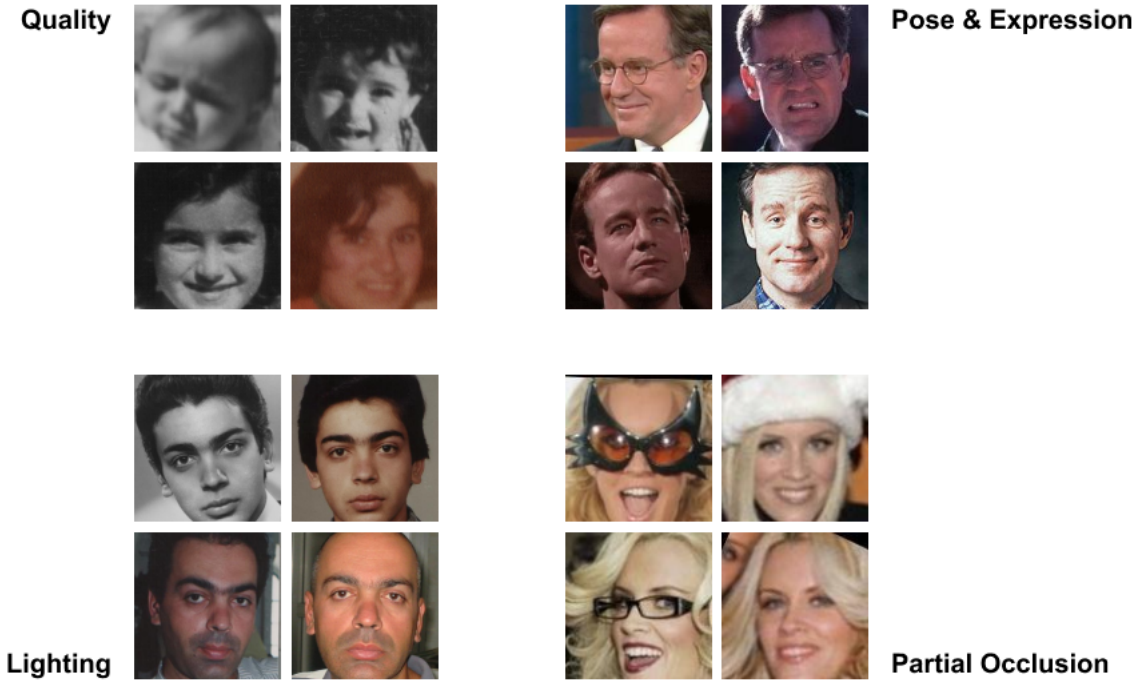

Figure 2.1: Challenges of Face Recognition with images from the FG-NET and CACD datasets.

In reality, numerous factors have an impact on Face Recognition including age variations, gender-specific characteristics, low resolution, pose differences, lighting variations, and facial expressions, as 2.1 portrays. The face image that is utilized directly affects how accurate face recognition algorithms are [MMS19]. Constructing a system akin to the human vision system is very challenging.

In the field of Face Recognition, there is the subtask of Age-Invariant Face Recognition. This subtask attempts to reduce how much the age variance factor impacts recognition. As mentioned in [Hua23], if the difference in age is broad, age variances may overpower the identity features in cross-age face identification, drastically reducing the recognition capacity. Even with Face Recognition's incredible success, algorithms still find it difficult to minimize the impact of age disparities in order to correctly identify people in a variety of real-world applications, including locating people who have been missing for a long time.

2.2 Age-Invariant Face Recognition (AIFR)

Age-Invariant Face Recognition (AIFR), is a Face Recognition method that can accomplish recognition invariant across various ages of a person. As mentioned in [Par19], the critical challenges are due to the wide variances in the facial images of an individual and the similarities between those of different individuals.

Age-Invariant Face Verification (AIFV) and Age-Invariant Face Recognition (AIFR) are the two main categories of Face Recognition across aging. AIFV is a binary-class classification issue, whereas AIFR is typically thought of as a multi-class problem. AIFV attempts to determine if two or more provided photos reveal the same or different identities, even with an age discrepancy. In contrast, AIFR is centered around matching a given photo of an individual with other photographs of the individual at different ages that are present in a collection.

One of the key topics of AIFR research is studying the manner in which aging impacts facial features. This subtask uses face aging datasets, detailed in 2.3, consisting of face images and other annotations, such as identity and age. The dataset is divided into several age groups [Par19].

In the childhood phase, between ages 0 to around 7, the person's identity is not mature, and their face characteristics aren't fully developed or established. Because of this, it can be difficult—even for the human eye—to match a child's face to that of their adult self [SM23]. Furthermore, the aging process of the face changes throughout life. Compared to face features, the texture of the skin alters less during infancy. The facial musculature gradually shifts during adolescence and then stabilizes. Up to the age of 50, the adult face gradually changes in structure. After that, more obvious aging symptoms like wrinkles and pigmentation changes start to show, accentuating these age-related characteristics.

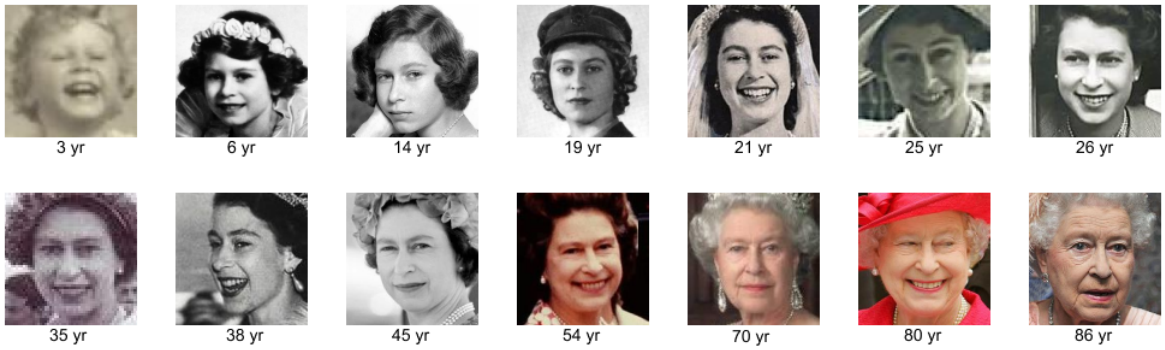

Figure 2.2: Images for illustrating face-by-age variations from the AgeDB dataset.

Figure 2.2 portrays an example of how a face can change from 3 years old to 86 years old and how different image variations can disguise face features. The human face is considerably altered by aging.

This aging process includes changes in subcutaneous fat, lessening bones, gravity, loss of facial volume, skin elasticity, and changes in skin texture. A person's work habits, diet, stress, health, smoking, substance abuse, and cosmetic surgery are all important environmental elements that contribute to a person's aging process. Furthermore, biological aspects should be taken into consideration as aging might manifest differently in various genders and ethnicities [Nay22]. Therefore, since facial aging affects each person differently, building a model that accurately detects the face characteristics of individuals at all age levels is challenging.

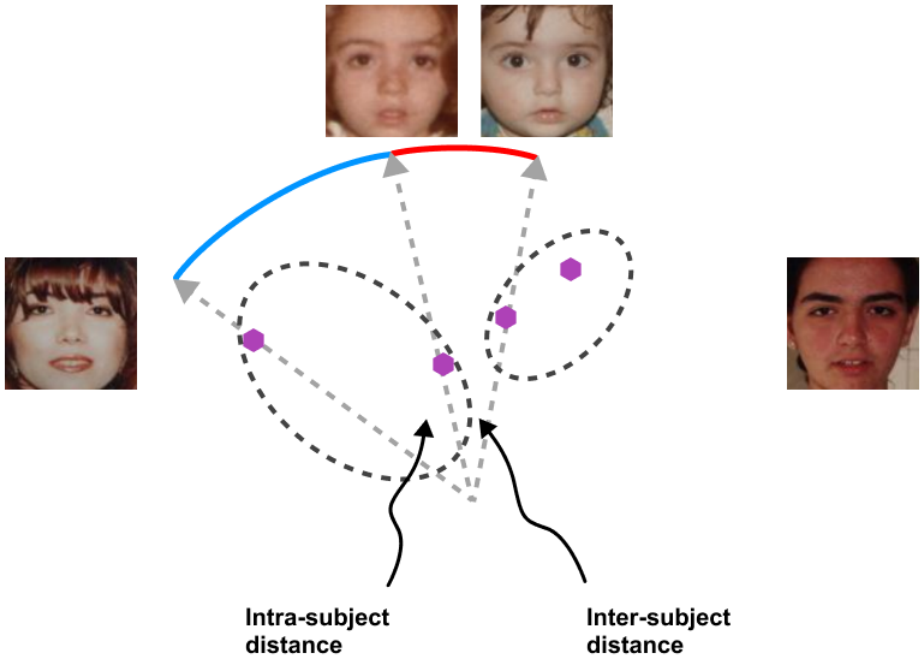

Figure 2.3: An illustration in which there are significant age differences, causing the intra-subject distance to be larger than the inter-subject distance from [HW19].

Pictured in 2.3, a further significant concern in this subtask is the alterations in the same person's photos as they age, or intra-subject variations, and the resemblance between the photos of different people, or inter-subject similarities. Because of this, many Face Recognition systems in use today struggle to recognize faces over large age differences. Age-Invariant Face Recognition has to consider the intra-subject variance, but also the inter-subject variance caused by age data.

As illustrated, face pictures are non-linearly separable and reside on large dimensions. Consequently, the process of classifying the facial photos becomes challenging. Any classification problem's primary goal is to extract the discriminative features, which a classifier employs for recognition and verification from the training datasets.

Still, research continues to improve both the recognition over aging accuracy and the identity preservation requirements.

Discussed in 2.3 are some of the currently available and helpful facial aging databases namely AgeDB [Mos17], FG-NET [Lan02], CACD [Che15], MORPH [Ric06], and VGGFace2 [QC18].

Also, detailed in [SM23] are the methods with which AIFR is typically tackled: Generative Methods, Discriminative Methods, and Deep Neural Networks (DNN) based Methods. Generative Methods work by identifying invariant characteristics from a given facial image to produce representations at various ages, which are then compared against test images. Discriminative Methods separate identity-specific information from the total facial information and make sure face recognition algorithms only use the essential identification information. This prioritizes the extraction of characteristics that hold true over aging. Additionally, [SM23] pointed out that DNN based Methods are used successfully and are mainly divided into feature extraction and classification tasks, using Discriminative Methods to improve the accuracy of age progression processes in facial recognition systems.

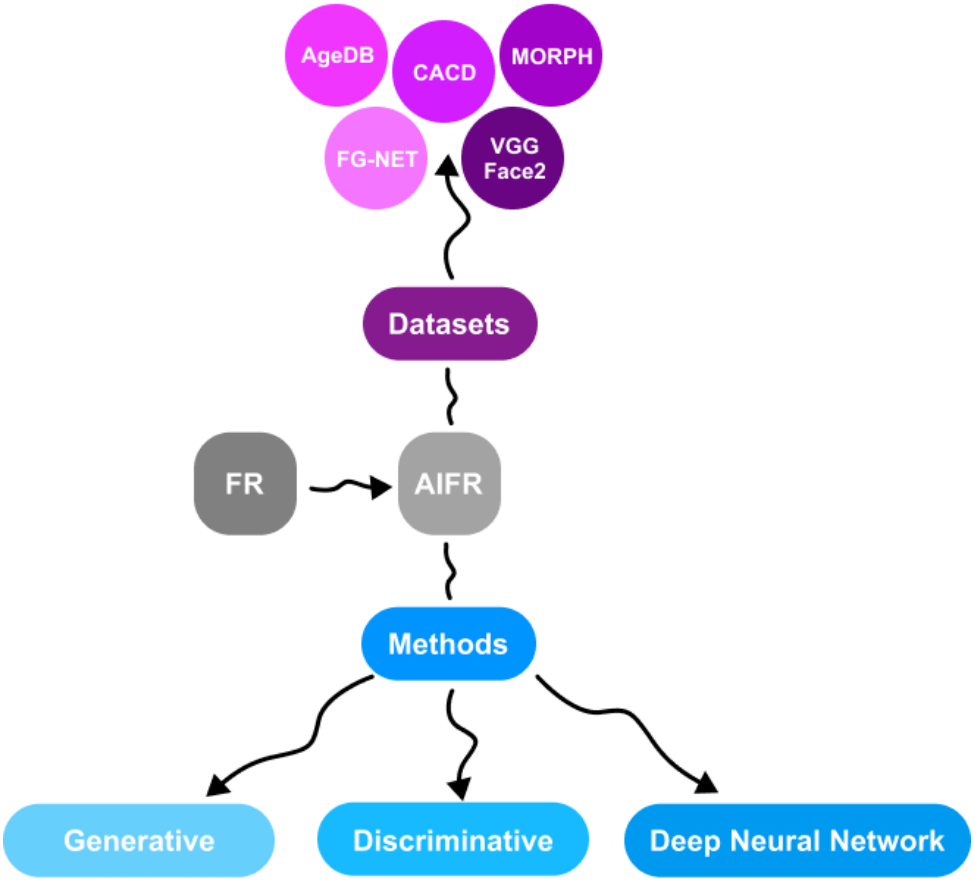

Figure 2.4: Simplified Age-Invariant Face Recognition taxonomy.

2.3 Datasets

In Age-Invariant Face Recognition research, datasets are essential. The ideal dataset would include multiple distinct face identities, a large number of photos of one's face at various ages with a small, but dense age gap between images, and annotations for the identity, age, and optionally gender or ethnicity. The most important datasets used for benchmarking this subtask are pictured in 2.4. We will present all of them and how they are particularly designed to address the recognition against aging problem.

Dataset	No. Images	No. Subjects	Age Labels	Age Range	Gender Labels
FG-NET	$1,002$	82	Yes	0 to 69	No
AgeDB	$16,488$	568	Yes	1 to 101	Yes
MORPH	$55,134$	$13,617$	Yes	16 to 77	Yes
CACD	$163,446$	$2,000$	Yes	16 to 62	Yes
VGGFace2	$3,310,000$	$9,131$	Yes	-	Yes

Table 2.1: Compact overview of the currently available aging databases.

2.3.1 FG-NET

The FG-NET dataset is relatively small compared to the other datasets we will present. It consists of a total of 1002 face images collected by mainly gathering age longitudinal photographs of the 82 unique subjects. The image number range per identity is 6 to 18. The minimum age is 0 and the maximum age is 69. The largest age gap is 45 years. Every face image has annotations on the attributes of age, and identity. Many face photos of people at young and elderly ages are included in the collection. First introduced in [Lan02], it was mainly created with the intention of investigating the issue of Age Regression and Progression in face images. It is used for AIFR, but also for Age Prediction.

2.3.2 AgeDB

The AgeDB dataset is a relatively big, manually collected, and in-the-wild dataset [Mos17]. It consists of a total of 16.488 face images collected from 568 unique subjects. For each individual, there are roughly 29 photos. The age range starts at 1 and goes up to 101. The average age range for each subject is 50.3 years. It consists of a wide range of personalities, such as writers, scientists, actors, and actresses. Every face image has annotations on the attributes of gender, age, and identity. Introduced in [Mos17], it guarantees an assessment free of noise of the different Face

Recognition algorithms since it was manually collected to ensure the precision of the age and gender annotations. AgeDB is utilized in AIFR and AIFV experiments and when the age difference between instances (pictures) of the same person rises, it is possible to quantify the ability to detect of the recognition algorithm.

2.3.3 CACD

The Cross-Age Celebrity Dataset [Che15], or CACD, is an extensive database of celebrity photos that were gathered from the Internet. Utilising the year (2004-2013) and the name of a celebrity as keywords, search engines were queried for photographs. It contains 163.446 face pictures from 2000 unique individuals. The range of ages is between 16 and 62 years. Every image has annotations on the attributes of identity, name, date of birth, and the approximate year when the image was shot. Introduced in [Che15], the dataset's diversity and scale provide a solid foundation for creating and testing algorithms in more realistic settings.

2.3.4 MORPH

One of the biggest longitudinal age progression databases is the MORPH dataset [Ric06]. There are two sets in total, Album 1 and 2. Album 1 is comparatively smaller than Album 2, with 1690 photos of the faces of 625 people ranging in age from 15 to 68. Album 2 contains 55.134 face photos of 13.617 individuals with an age range from 16 to 77 years. The dataset stands out in particular for having excellent photos and information that includes annotations about age, gender, and race [Ric06]. This dataset is not publicly available at the moment, but it is available for commercial purchase.

2.3.5 VGGFace2

Another extensive face collection is the VGGFace2 dataset [QC18]. It is made up of over 3.310.000 pictures collected from 9131 unique individuals. For each individual, there are roughly 362.6 photos. Introduced in [QC18] the authors concentrated on achieving a very low label noise and a high pose and age variety while creating the dataset, which makes VGGFace2 a good option for training cutting-edge deep learning models on challenges involving AIFR. This dataset used to be accessible to the general public, but at this time it is not.

2.4 Methods and Related Work

A major scientific difficulty in Computer Vision is addressed by the advancement of Age-Invariant Face Recognition. These include the necessity for models that can generalize successfully from limited age-specific training data, the scarcity of longitudinal face data, and the heterogeneity in aging processes among various factors. Consequently, researchers and developers have investigated several strategies to address this issue, all with the goal of improving the precision and dependability of face recognition systems for a vast range of age groups.

Pictured in 2.4 are the three methods, mentioned previously, that have been recently used to address the AIFR subtask: Generative Methods, Discriminative Methods, and Deep Neural Network (DNN) based Methods. All techniques are used to train the models to learn (or deduce) important traits that are constant with age. The test photos are then identified using these features. Generative Methods, 2.4.1, focus on simulating the aging process of the face. Discriminative Methods, 2.4.2, focus on identifying characteristics that are both age-invariant and sufficiently unique to distinguish between people. Age-Invariant Face Recognition has been transformed by Deep Neural Network (DNN) Based Methods, 2.4.3, which make use of large-scale data processing power and the capacity to learn unique, hierarchical characteristics from data. These techniques have evolved throughout the years in response to continuous efforts to develop Age-Invariant Face Recognition systems that are more resilient, flexible, and effective.

2.4.1 Generative Methods (GM)

In Generative Methods (GM), aging characteristics and personal identity features are combined to create 2D or 3D face models. In Face Matching or Age Estimation, aging models may compensate for the aging process [SM23]. This approach is frequently used to solve the Age Estimation task. However, researchers have also applied it to classification issues related to face aging.

This model often works with two underlying models at a particular age: texture aging and shape aging. Several age groups' texture and shape features are taken from the training datasets, and various approaches employ these features to fit the parameters of the texture and shape models. The model parameter error will be identified by comparing the outcomes of the matching procedure. In order for the model to converge in accordance with the penalty function, the procedure will be repeated up to a certain threshold error value.

[JW06] developed an age simulation that established a mathematical connection among the texture and form facial embeddings and their corresponding number of

years. This allowed them to alter the facial shape and texture from an initial age to a desired target age.

In a different approach, [Par08] research introduced a three-dimensional model that captures aging patterns in both face texture and form, by using a three-dimensional model that was morphable to simulate aging on two-dimensional facial images.

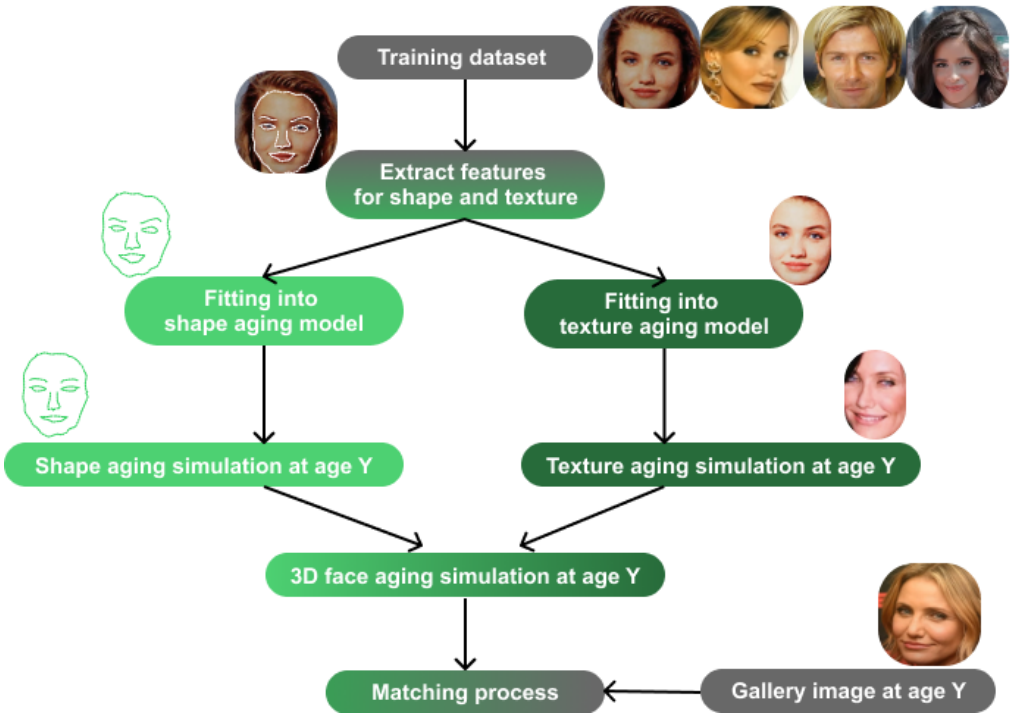

Figure 2.5: Process of Generative models inspired from [SM23].

In Generative modeling, handling variations becomes increasingly complex as the datasets grow. These datasets not only cover a range of ages but also include variations due to different facial expressions, postures, and lighting conditions. This complexity presents significant challenges in maintaining consistency across such multidimensional changes. The selection of parameters in a Generative model is frequently not successful in modeling the entire range of variations. In an ideal scenario, Generative methods would simulate the distribution of distinct classes, that is, pictures that are specific to one person.

Generative models are better with smaller training datasets [SM23]. However, utilizing Generative approaches for generating facial images for ages that are not captured in a dataset is still a noteworthy and promising application field of the Generative Methods.

2.4.2 Discriminative Methods (DM)

In contrast to Generative Methods, Discriminative Methods (DM), don't depend on face modeling. Robust feature descriptors and discriminating learning or classification techniques are the foundation of this method. In general, Discriminative Methods need less computing power and yield optimal outcomes rapidly. When there is a lot of training data, the model performs more accurately than the Generative models because Discriminative models converge to fewer asymptotic errors. Therefore, for these models, the error rate reduces with increasing training sample count. A variety of categorization techniques may be applied to match the image with the right class. Learning the low-level feature space is the initial phase for this Age-Invariant Face Recognition method.

To generate a powerful discriminative feature space, researchers often utilize many features. In order to learn the joint discriminative feature space, features are combined with a few fusion approaches. In order to learn appropriate low-level feature spaces, pictures used to train are transmitted onto suitable feature spaces. Additionally, this redundancy would be eliminated, maybe using a random sampling technique, in order to maximize the usage of memory storage and processing time. After extracting facial characteristics, a classifier categorizes the training data using these unique features per class, and the error is calculated by contrasting the extracted label with the actual output label.

Cross Age Reference Coding is a novel coding technique that was introduced by [Che15] using a knowledge-based methodology for encoding the local characteristics of a photo into an n-dimensional feature space using a reference set of photographs of various individuals. Using this technique, Age-Invariant representations of local characteristics are created, allowing for consistent traits between two images of one single individual at distinct years in a new comparison scenario. Consequently, this method achieves substantial effectiveness in identifying and retrieving faces over various age groups, with a recognition rate of 92.8% on the MORPH dataset.

Discriminative Methods are especially valuable since they prioritize stability over variability and directly address the problem of telling apart individuals within the same age group as well as identifying individuals at various ages.

2.4.3 Deep Neural Network (DNN) Based Methods

Deep Neural Networks (DNN) based Methods, are increasingly utilized in the context of AIFR, with a special emphasis on Convolutional Neural Networks. In 2.7 we have an architecture of every step that is taken in a standard CNN. The loss function is an essential component of CNN's design as it gauges the algorithm's

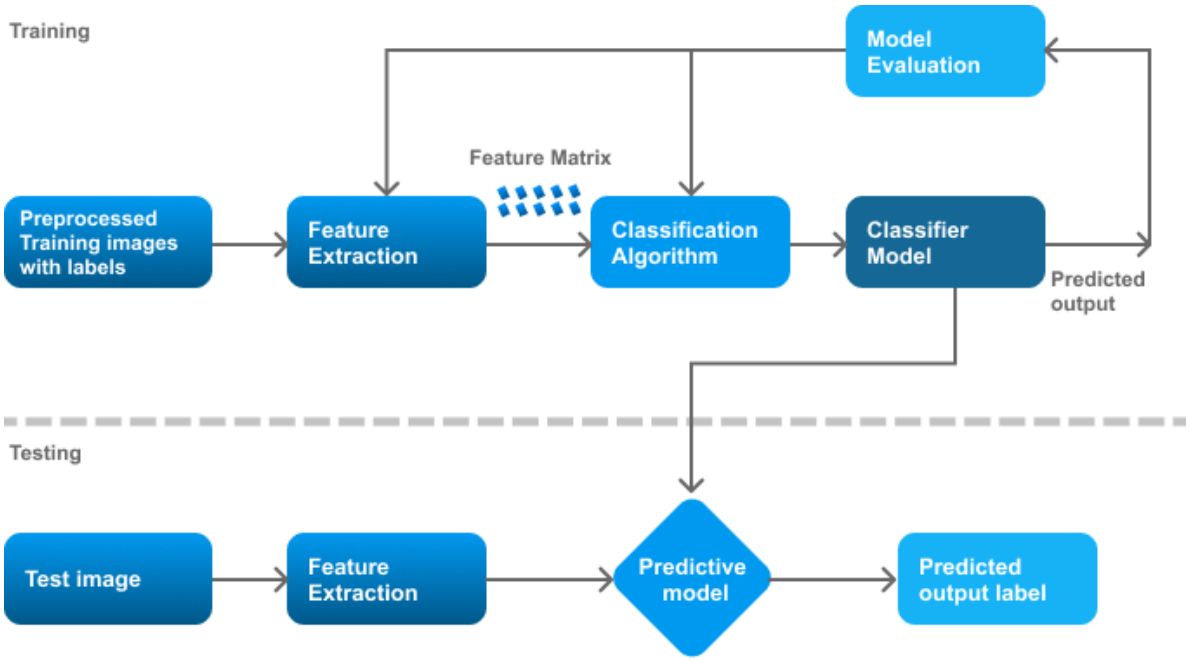

Figure 2.6: Process of Discriminative models inspired from [SM23].

performance. The degree to which the algorithm identifies, classifies, or fits the test sample. The error is calculated using the loss function, and the CNN's parameters and weights are then changed to point toward the ideal feature subspace.

Shallow Age-Invariant Face Learning, coined by [Isl21], involves grouping pairs of images with a broad age gap and applying State-Of-The-Art (SOTA) methods for face verification. This study assessed the performance of 6 prominent deep face models, including Open-Face, VGG-Face, Face-Net, Deep-ID, DeepFace, and Arc-Face, across different metric learning techniques [Isl21]. The three cross-age data sets were rearranged into positive and negative pairings with various age ranges. According to [Isl21] research and testing, Arc-Face and Face-Net perform better for shallow AIFR issues than other SOTA. The majority of Face Recognition algorithms degenerate for shallow AIFR and overfit in terms of feature dimensions. Thus, matching picture pairings across a wide age range remains a challenging problem within the domain of Face Recognition.

[HW19] factorized the facial features into two uncorrelated parts, age-related and identity-related components, through a deep feature factorization framework. They have proposed a Decorrelated Adversial Learning (DAL) method to minimize the correlation between decoupled features of age and the identity of a similar person. The recognition rate achieved on FG-NET was 94.5%, and on MORPH-II was 98.93%.

The linear factorization module of [HW19] was enhanced by [Hua23]. The identity-related component, which lacks facial spatial information, is formed in a one-dimensional

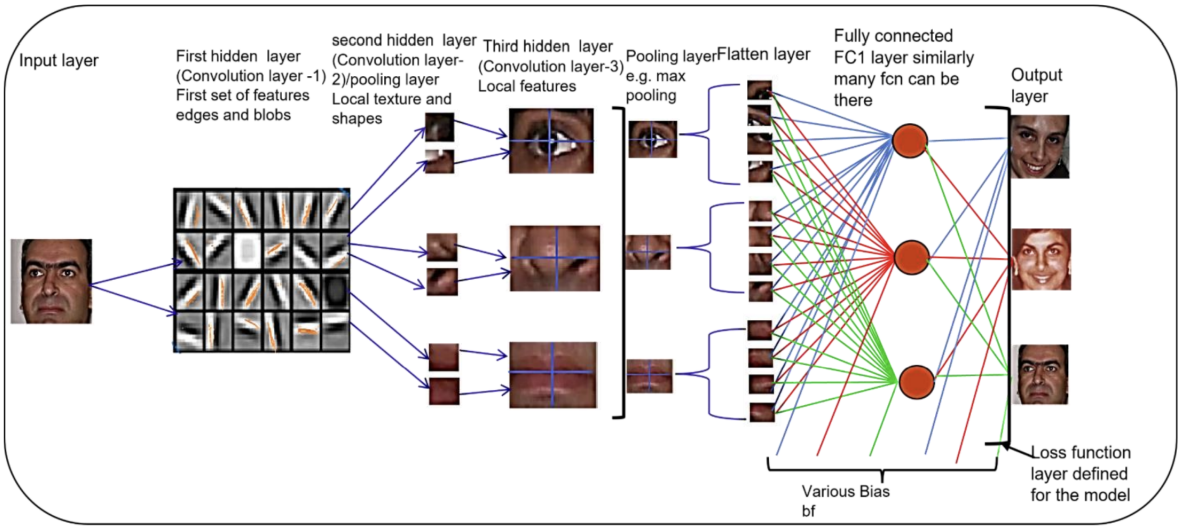

Figure 2.7: Deep Neural Network's feature extraction from [SM23].

embedding vector. As a result, in high-dimensional feature space, they suggested an attendance-based features decomposition that has a stronger semantic correlation. The model is trained using GAN. On FG-NET this model yielded a 94.78% recognition accuracy.

DNN Based Methods have demonstrated a remarkable capacity to learn characteristics from the raw pixels that are both age-invariant and age-specific. Large datasets of face images, frequently enhanced with digitally aged or rejuvenated images, are used to train these networks. These techniques are important because they offer better accuracy and scalability.

2.5 Issues and Challenges

Age-invariant Face Recognition has seen significant advancements, but there are still a number of major problems that make it difficult to create and implement efficient systems. These difficulties result from the intricacy of face aging on its own, as well as the technological constraints of recognition algorithms.

Individuals experience aging in different ways and are subject to a variety of variables including genetics, habits, health, and environment. Because of this variety, it is challenging to forecast and simulate how each person's facial characteristics will evolve over time. Such varied aging trajectories are difficult for current Generative models to account for, which can result in errors in the modeling of aging faces and the identification of people at different ages.

Large datasets covering a broad range of ages for each person are necessary for reliable Age-Invariant Face Recognition, ideally capturing the progressive changes in their facial characteristics over time. These longitudinal datasets are uncommon and costly to gather, though. The majority of the currently available datasets like FG-NET, AgeDB, CACD, MORPH, and VGGFace2, have the following issues: an imbalanced number of age-progressive images per person or large age gaps, and they include variations in face images such as illumination, clarity, or facial expressions. These factors complicate the training process and threaten achieving optimal results.

Furthermore, the majority of datasets that are currently accessible have biases towards specific demographics, usually towards younger. When used in varied, real-world settings, recognition algorithms may perform worse due to this lack of diversity as a person's unique age patterns are based on numerous factors including genetics, nutrition, circumstance, origins, and so on.

Overcoming these obstacles and progressing in Age-Invariant Face Recognition requires ongoing study, varied data collecting, and rigorous testing conditions.

Based on the premise that the decomposed components accurately reflect face information, this survey points to the importance of feature decomposition in invariant feature learning. Nonetheless, the identity characteristics could contain details like gender or age, and the dissected components could have latent relationships with one another. However, being able to classify the decomposed components correctly and the identity-dependent component being rich in information is a step towards different feature invariant Face Recognition.

Chapter 3

Theoretical Foundations

3.1 Convolutional Neural Networks (CNN)

Within the field of Artificial Intelligence, Machine Learning revolves around constructing mathematical algorithms and models that eliminate the need for manual instructions for computers to do particular operations. Rather, these systems use patterns and inferences drawn from data to learn and make decisions. Accuracy is determined by calculating how closely the selected actions match the proper ones.

Deep Learning is an advanced branch of Machine Learning that models complex patterns and decision-making processes using different Neural Networks (NNs). Because of this Deep learning is used in this research. A range of methods are used in Deep Learning, such as Supervised Learning, a technique where the model learns from data that has labels, or Unsupervised Learning, which looks for patterns in unlabeled data, and Reinforcement Learning, where an agent learns to make decisions by receiving rewards for actions. While there are several types of methods, classification-based Supervised Learning is used in this research because training a model on a labeled dataset, where each input (an image of a face) is associated with multiple labels (the identity, age, and gender of the face in the image) is essential for teaching the model to distinguish between different individuals' face, age, and gender effectively.

A Neural Network is composed of interconnected nodes or neurons. These are usually aggregated into layers and linked by edges. Every layer involves a linear transformation, where the weights of a matrix are learned. Different layers have various ways of changing the information they receive. Signals may flow via a number of intermediate levels, often referred to as hidden layers, between the first layer and the last layer, referred to as input layer, and output layer respectively. After receiving input signals from the previous layer, each layer's neurons apply a set of weights and biases to create an output signal, which is then passed on to the next

layer. If a network contains two or more hidden layers, it is commonly referred to as a Deep Neural Network.

The multi-layered structure of the human visual brain served as the model for Convolutional Neural Networks. The visual cortex and CNNs both have a hierarchical structure, with deeper layers containing more complex characteristics built up from simpler features retrieved in earlier levels. This makes it possible to represent visual inputs in ever more complex ways. Rather than connecting to the full visual field, neurons in the visual cortex only make connections to a limited area of the input. Similar to this, the convolution procedure in a CNN layer only connects the neurons to a limited area of the input volume. Efficiency is facilitated by this local connectivity.

A Convolutional Neural Network is primarily composed of 4 main parts.

Convolutional layers are the most important structures of a CNN. The primary mathematical operation carried out is Convolution, the application of a sliding window function to a matrix of pixels representing an image. The terms ”filter” or ”kernel” are used to refer to the sliding function that is applied to the matrix. Several equal-sized filters are applied in this layer, and each filter is utilized to identify a certain pattern from the picture, including the edges, curves, or overall shapes of the digits. The weights of the kernels are determined during the training procedure of the Neural Network. Several distinct feature maps are extracted at every level of visual processing. CNNs replicate this by using different filter maps in every convolution layer.

Pooling Layers are used to pull the most significant features from the convoluted matrix. Features are detected by visual cortex neurons independent of where they are in the visual field. By summarising local information, CNN pooling layers offer a certain level of translation invariance. This is accomplished by using a few aggregation processes, which decrease the feature map's (convoluted matrix) dimension and, as a result, the amount of memory needed for network training.

The most often used aggregating functions that are available are **Max pooling** (the feature map's maximum value), **Sum pooling** (equivalent to adding up each feature map value), and **Average pooling** (average of all the values). The feature map is flattened by the last pooling layer so that the fully linked layer can analyze it.

Fully Connected Layers are the last layers of the Convolutional Neural Network, and their inputs match the one-dimensional matrix that the last Pooling Layer produced, which has been flattened.

Activation functions are used to increase the non-linearity of the network [Upr22]. One of them is the **Rectified Linear Unit**, or**ReLU**, is an activation function performed after every convolution process. The response characteristics of neurons in

the visual cortex are nonlinear, this activation function gives a CNN its non-linearity. By teaching the network non-linear correlations between the image's characteristics, this function strengthens the network's ability to recognize various patterns. It also aids in lessening the issues with vanishing gradients. Other activation functions are the **Sigmoid**, **Exponential linear unit (ELU)**, **Hyperbolic tangent (tanh)**.

3.2 Training a Neural Network

To achieve efficient learning, a neural network must be trained by carefully adjusting a variety of hyperparameters and optimization strategies.

Gradient descent is an optimization procedure that iteratively moves towards the steepest descent, as indicated by the function's negative gradient, in order to discover the minimum of a function. Starting from an initialized position, the procedure proceeds in steps that are proportionate to the function's negative gradient at that point. The size of the steps is determined by the step size, referred to as the learning rate. The algorithm modifies the steps forward or backward to obtain the local or global minimum based on the gradient at each subsequent step.

Batch gradient descent processes portions of the data at each learning step instead of the full dataset in an effort to minimize computing load. This method updates the model parameters by first computing and summing gradients over a batch of samples. Compared to using the complete dataset, this method guarantees smoother convergence and aids in memory management. Learning rate decay can be avoided when batch gradient descent is used, allowing for the maintenance of a fixed learning rate. The amount of samples processed in a single iteration is determined by the batch size, which is an important factor. Greater batch sizes impact the model's convergence behavior and demand more memory.

Stochastic Gradient Descent (SGD) is a variation of gradient descent where the model parameters are updated using the gradient of the loss function evaluated on a single sample or a mini-batch of samples, rather than the entire dataset. This approach introduces noise into the optimization process, which can help escape local minima and lead to faster convergence. The stochastic nature of SGD can make the optimization process more volatile, but it often results in finding better minima compared to batch gradient descent.

Momentum helps accelerate gradient vectors in the right directions, thus leading to faster converging. It can help to avoid getting stuck in local minima by smoothing out the updates. The momentum parameter usually takes values between 0 and 1.

The learning rate determines the size of the steps the optimizer takes when adjusting the model parameters. A learning rate that is too high can cause the model to converge too quickly to a suboptimal solution, or even diverge, while a learning

rate that is too low can result in a prolonged training process. Finding the optimal learning rate often involves experimentation and may include using techniques like learning rate scheduling or adaptive learning rates.

Batch size refers to the number of training samples processed before the model's internal parameters are updated. Larger batch sizes can lead to more accurate estimates of the gradient but require more memory and computational power. Smaller batch sizes make the training process noisier and can lead to faster convergence, albeit with less accurate gradient estimates.

Number of epochs refers to the number of complete passes through the entire training dataset. More epochs allow the model to learn better but can lead to overfitting if training is prolonged excessively. The optimal number of epochs is typically determined through validation testing.

Overfitting occurs when a model becomes overly adept at recognizing patterns and anomalies in the training set. A model that learns this way does poorly on fresh, unknown data but well on training data. Convolutional neural networks (CNNs), in particular, are particularly prone to overfitting because of their tremendous complexity and potential to discover intricate patterns in vast amounts of data.

Overfitting in CNNs may be reduced by using a variety of approaches.

Dropout represents the process where some neurons are arbitrarily removed in the training phase, forcing the remaining neurons to pick up new characteristics from the input data.

Batch normalization is the method of normalizing the input layer by varying and scaling the activations, the overfitting is somewhat mitigated. This method is also applied to stabilize and expedite the training process.

Early stopping involves tracking the model's performance on validation data during the training phase and halting the training as soon as the validation error stops improving.

Noise injection is used to strengthen the model and keep it from weakly generalizing, noise is introduced to the hidden layers' inputs or outputs during training.

L1 and L2 normalization represents the process in which the loss function is penalized depending on the magnitude of the weights using both L1 and L2. More precisely, L1 promotes the sparing of the weights, which improves feature selection. L2, also known as weight decay, on the other hand, promotes tiny weights in order to limit their impact on the predictions.

Data augmentation is the method of adding random modifications, such as rotation, scaling, flipping, or cropping, to the training dataset in order to artificially increase its size and variety.

Chapter 4

AIFR Research Approach and Individual Contributions

4.1 Backbone

4.1.1 Backbone Custom CNN

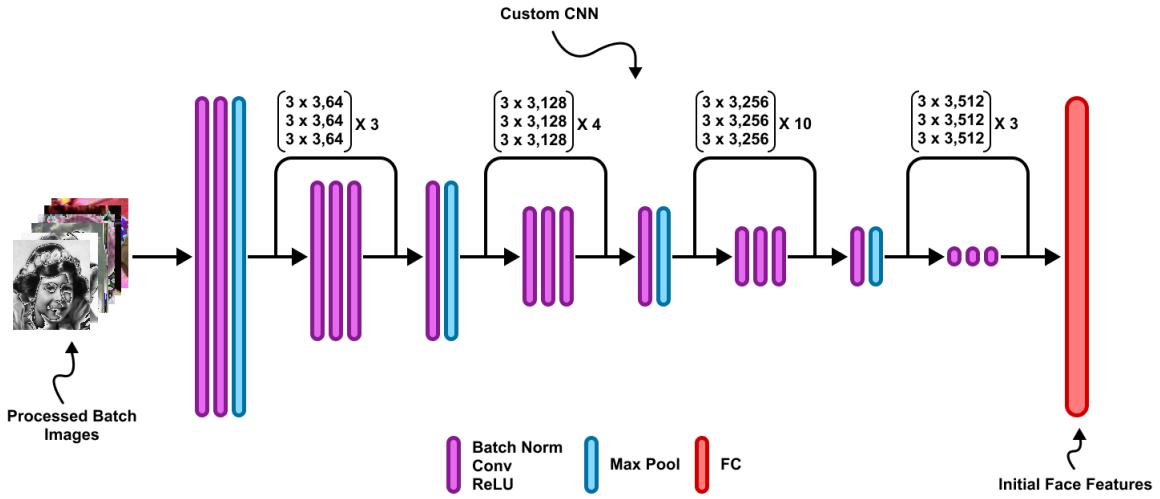

Figure 4.1: The Backbone Custom CNN's architecture.

In the first stages of developing the Age-Invariant Face Recognition model, we used as a Backbone a Custom CNN inspired by the ResNet architecture. The idea behind this Backbone is from [HW19], with an addition to the Orthogonal Embedding CNN (OECNN) from [YWZ18] because of the added Batch Normalization.

The Custom CNN's architecture is pictured in Figure 4.1. It is a 64-layer CNN, consisting of 4 stages with respectively 3, 4, 10, and 3 stacked Residual Blocks and a final Fully Connected (FC) layer that outputs the initial face features of 512 dimensions. A Residual Block consists of 3 stacked units of 3x3 Batch Normalization,

Convolution, and ReLU layers.

In order to implement this we used Pytorch. A class named BackboneCNN is initiated with the possibility of specifying a weight initializer and dropout probability. In order to construct the layers, we first use a Singleton class that creates an instance of the layers configuration of the Backbone. Then this instance is used in the Builder class that constructs the layers of the Backbone. There are 3 different types of layers MaxPool, BatchNormConvReLU, and ResidualBlock containing 3 stacked units of BatchNormConvReLU. Each layer is added based on the provided configuration presented above in the architecture. The final layer is composed of Batch Normalization, Flatten, Dropout layers, and the Fully Connected Layer with Batch Normalization.

The results we achieved using this as a Backbone were insignificant, as the dataset's size and diversity were limited, and the task it had to learn was immensely dependent on these resources. Because of this, we moved on to Transfer Learning.

4.1.2 Backbone Inception-Resnet-v1 via Transfer Learning

Developing a Deep Neural Network from the ground up demands a lot of time and typically requires extensive datasets to obtain satisfactory accuracy [SM23].

To train a supervised model a large amount of processing power and labeled datasets are needed. This contributed to the rise of Transfer Learning - the use of models that have already been trained [SM23]. The explanation is that Transfer Learning may be thought of as a remedy for the requirement for large training datasets, which were essential for training Neural Networks to provide results that are significant.

The activation layers of a pre-trained network serve effectively as feature extractors, allowing for the application of various classification algorithms. Renowned examples of such models include VGG16, VGG19, ResNet50, InceptionV4, Inception-Resnet, and Xception. These architectures vary widely in terms of Convolution layers, Pooling layers, Fully Connected layers, or hyperparameters. Widely available across various platforms, these models are found in popular libraries like Pytorch, Tensorflow, and Keras, among others.

Because the dataset and resources we had were limited, we choose to use Transfer Learning. We conducted research for design and performance on different architectures from above and concluded that the best suited for our task is the Inception-Resnet-V1 CNN introduced in [CS16] and pre-trained by [Esl17] on the VGGFace2 [QC18] dataset. This serves as the feature extractor, converting input images into embeddings or initial face features.

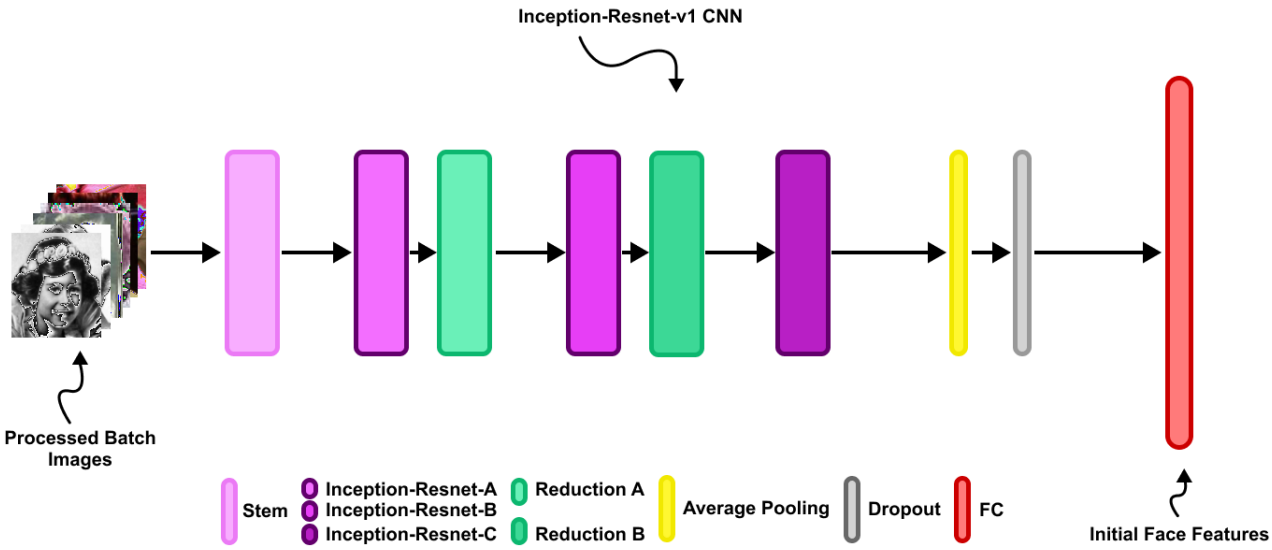

Figure 4.2: The Backbone Inception-Resnet-V1's architecture.

As described in [CS16], the **Inception-Resnet-v1** architecture pictured in 4.2 combines the strengths of two powerful architectures: the Inception modules provide a mixed level of convolutions to capture information at various scales, while the residual connections facilitate training of deep networks by allowing gradients to flow through the network layers without diminishing.

The architecture introduced by [CS16] begins with a **Stem** block that performs initial convolutions to reduce dimensionality before further processing. This block combines convolutions of varying sizes to capture a broad spectrum of features from the input images, setting the stage for deeper analysis in subsequent layers.

Following the **Stem**, [CS16] used several Inception blocks: **Inception-Resnet-A**, **Inception-Resnet-B**, and **Inception-Resnet-C**. Each of these blocks is designed to handle different aspects of the image:

- **Inception-Resnet-A** blocks focus on capturing smaller details with a combination of smaller convolutions.
- **Inception-Resnet-B** blocks apply larger convolutions to abstract higher-level features that are more spatially spread out.
- **Inception-Resnet-C** blocks, positioned deeper within the network, optimize the feature channels, enhancing the high-level features captured in previous blocks.

To reduce dimensionality and prepare the data for the next stages, [CS16] used two reduction blocks, **Reduction A** and **Reduction B**, strategically placed in the

network. These blocks perform down-sampling, which helps in reducing computational complexity and focuses the network on the most salient features.

Towards the end of the network, [CS16] append an **Average Pooling** layer, a **Dropout** layer, and a **Fully Connected (FC)** layer to the network.

- **Average Pooling** layer helps to reduce the spatial dimensions while preserving the most critical information.
- **Dropout layer** mitigates overfitting by randomly dropping units during the training phase, ensuring that the model generalizes well to unseen data. We used a **0.4** dropout probability for all models.
- **FC** layer consolidates the learned features into a final embedding vector that represents the initial face features.

These components collectively enable the **Inception-Resnet-v1** to effectively learn and generalize from a diverse set of facial images, making it an excellent choice for our Age-Invariant Face Recognition models.

4.2 Baseline Single-Task Model

In order to see how well our Multi-Task Model for AIFR improves a simple FR model, we first started developing our Baseline with a Single-Task Model that only classified the identity of the person.

4.2.1 Loss Function

The extraction of features from faces so they may be compared to every other face in the collection is the first stage in the Face Recognition process. Similar to [HW19], to supervise the learning of identity we choose to use a margin loss.

The **Large Margin Cosine Loss**, or **CosFace**, modifies the traditional softmax loss into a cosine loss by normalizing both the features and the weight vectors, as detailed in [HW18]. This modification introduces a cosine margin to enhance the decision boundary between classes. Thus, we obtain the lowest intra-class margin and the maximum inter-class margin. The loss function is mathematically represented as:

$$\mathcal{L}_{ID} = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{e^{s(\cos(\theta_{y_i,i})-m)}}{e^{s(\cos(\theta_{y_i,i})-m)}+\sum_{j\neq y_i}e^{s\cos(\theta_{j,i})}} \tag{4.1}$$

where N is the number of identities, y_i is the true class label for the i-th sample, $\theta_{y_i,i}$ is the angle between the feature vector of the i-th sample and the weight vector corresponding to its true class, $\theta_{j,i}$ is the angle between the feature vector of the i-th

sample and the weight vector of any other class j, s is a scaling factor that controls the sharpness of the decision boundary, m denotes the margin that separates identities further by increasing the inter-class variance and decreasing the intra-class variance.

The **numerator** of the fraction inside the logarithm function represents the exponential score for the correct class, adjusted by subtracting the margin m. This subtraction aims to push the features of the correct class further away from the decision boundary, thus enhancing the discriminative power of the model.

The **denominator** sums over all class scores, with the score of the correct class being adjusted by the margin, while the scores for other classes are not. This structure ensures that the model learns to widen the cosine margin between the correct class and all other classes. Such an adjustment improves the separation between classes, making the classifier more robust to variations within facial appearances and across different identities.

The CosFace loss aims to optimize the decision boundaries between the classes by adjusting the scale s and margin m, making the classifier robust to variations within the classes. We used a scale of **32** and a margin of **0.1** for all models.

4.2.2 Identity Classification

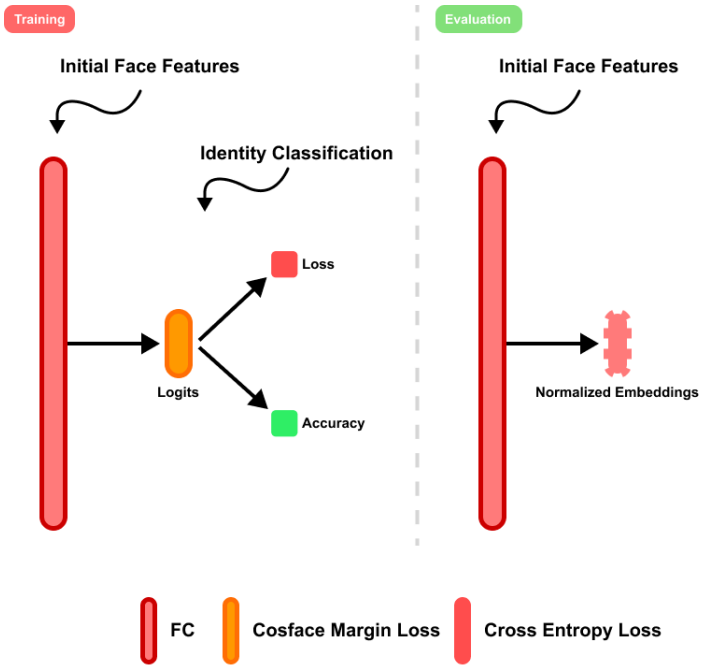

Figure 4.3: How are the embeddings from the Backbone transformed for the Identity Classification.

For the classification process, we use a Softmax layer combined with Cross Entropy Loss. This setup effectively maps the raw, unnormalized embeddings pro-

duced by the Backbone into a probability distribution over the target classes. The transformation from embeddings to logits, and subsequently to normalized probabilities, facilitates a more straightforward comparison against true class labels.

The Softmax function is pivotal in converting the logits (output from the last neural network layer prior to the output layer, without normalization) into probabilities by exponentiating and normalizing each output. It is mathematically defined as:

$$\sigma(\mathbf{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{N} e^{z_j}} \tag{4.2}$$

where z_i represents the logit corresponding to the i-th class, and N is the total number of classes.

The Cross Entropy Loss measures the performance of the classification model. Cross entropy loss increases as the predicted probability diverges from the actual label. It is defined for a single sample and a single class as:

$$\mathcal{L}_{CE} = -\sum_{i=1}^{N} y_i \log(p_i) \tag{4.3}$$

where y_i is the binary indicator if class label i is the correct classification for the observation, and p_i is the predicted probability of the observation of class i.

Since we developed the model using Pytorch, the off-the-shelf Cross Entropy Loss function, applies the Softmax function internally.

The forward method of this classification model involves the following steps:

Algorithm 1 Forward Method for Classification Model

Require: Pre-processed images in batches, Ground truth labels
1: **if** training mode **then**
2: Extract features from images using Backbone
3: Process features with CosFace margin loss to compute logits
4: Apply Cross Entropy Loss function to convert logits to probability scores
5: Calculate loss between predicted probabilities and actual labels
6: **else if** evaluation mode **then**
7: Extract features from images using Backbone
8: Output normalized embeddings for evaluation purposes
9: **end if**

This method not only classifies but also evaluates the embeddings' quality by measuring how well they correspond to actual class labels, thereby assessing the model's performance during both training and evaluation phases.

4.3 Multi-Task Model

In the pursuit of a robust AIFR system, we continued our study by developing a Multi-Task Model capable of simultaneously classifying three key features from facial images: identity, age, and gender. This model integrates a Feature Residual Factorization Module (FRFM) and three dedicated discriminators or classifiers, one for each attribute, facilitating a joint supervised learning approach that optimizes the feature extraction process for enhanced discriminative performance.

4.3.1 Feature Residual Factorization Module

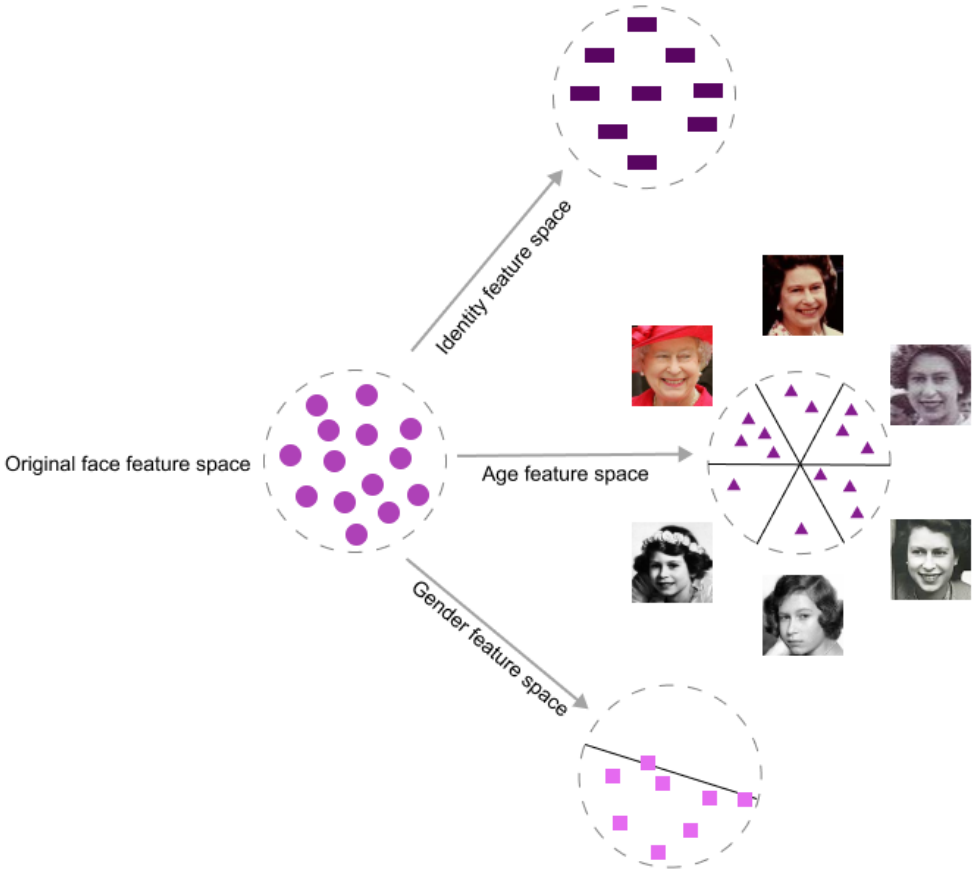

Figure 4.4: The original face features are decomposed into the age features, the gender features, and the identity features.

Drawing inspiration from advanced factorization techniques in Deep Neural Networks [HW19, KC18], our Feature Residual Factorization Module (FRFM) separates combined facial embeddings into three independent parts: age features (x_{age}), gender features (x_{gender}), and identity features (x_{id}). The disentanglement of the face features is illustrated in 4.4. This separation is crucial for handling the distinct aspects of facial analysis and is achieved through a linear decomposition approach.

This factorization is done through the Feature Residual Factorization Module similar to [KC18]. Given the embeddings x that the backbone CNN B extracts from an input picture p, the linear decomposition is:

$$x = x_{\text{age}} + x_{\text{gender}} + x_{\text{id}}, \tag{4.4}$$

where x_{id} denotes the identity-reliant features, x_{age} denotes the age-reliant features, and x_{gender} denotes the gender-reliant features.

The age and gender features are encoded through a residual mapping function $x_{\text{age}} + x_{\text{gender}} = R(x)$ and $x = x_{\text{id}} + R(x)$. Using the Feature Residual Factorization Module, we obtain age-reliant and gender-reliant features, and the residual part from the initial face feature vector are the identity-reliant features:

$$x_{\text{age}} + x_{\text{gender}} = R(x), \qquad x_{\text{id}} = x - R(x) = x - x_{\text{age}} - x_{\text{gender}} \tag{4.5}$$

As pictured in 4.5, the distinct pathways for age, gender, and identity within the FRFM not only ensure targeted feature enhancement but also mitigate the interference between these attributes during the learning process. This methodical separation aids in reducing the model's complexity and enhances the interpretability of learned features, essential for applications requiring transparent decision-making processes. By structuring the learning process to simultaneously optimize for multiple attributes, the model not only becomes more efficient but also mirrors a more holistic form of human-like perception, where multiple cues are used to form a comprehensive understanding of the observed subject.

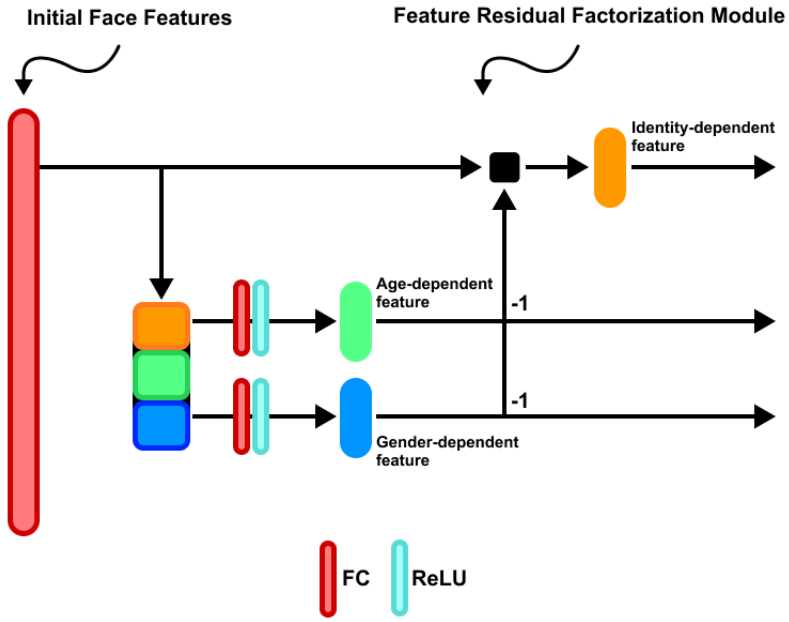

Figure 4.5: The Feature Residual Factorization Module (FRFM) architecture.

4.3.2 Identity Discriminator

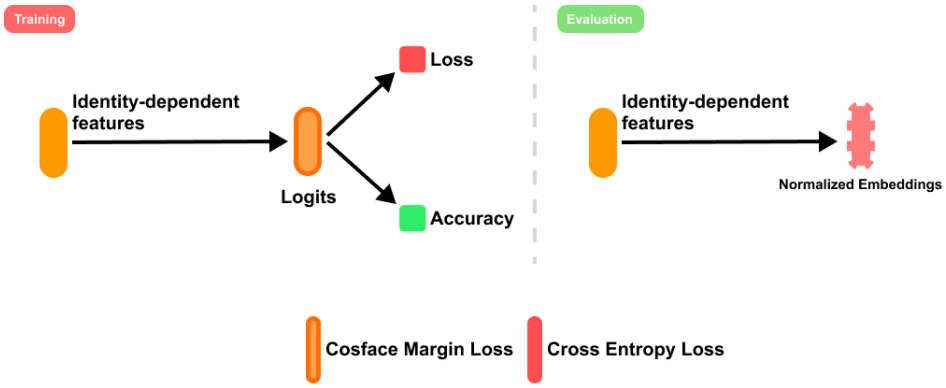

Figure 4.6: Identity Discriminator architecture.

The Identity Discriminator is a crucial component of our Multi-Task Model, designed specifically to recognize and verify individual identities based on extracted facial features. This module employs an architecture that mirrors the effective strategies used in our Single-Task model, detailed in 4.2.2, but only on the discovered identity features.

4.3.3 Age Discriminator

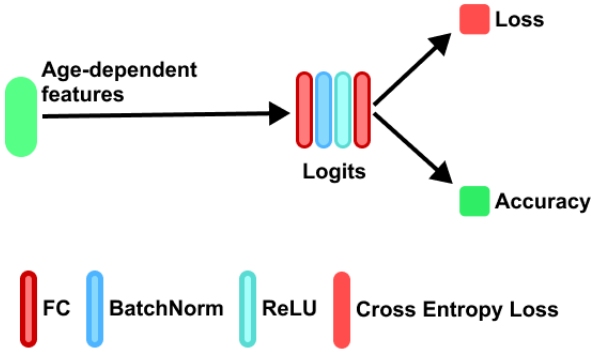

Figure 4.7: Age Discriminator architecture.

To refine the expression of age-related attributes, the variable x_{age} is passed into the Age Discriminator, ensuring the learning of age-specific information. Our data contains age labels, but since they may have uncertain noises in practice, classification on age groups is performed. Three age groups are defined as follows: $[0-25]$, representing youth; $[25-55]$, denoting middle-aged individuals; and $[56+]$, for the elderly subjects. We stack one Fully Connected Layer, Batch Normalization, ReLU activation, and another Fully Connected Layer over the x_{age} features to calculate the logits, then use the Cross-Entropy Loss for performing the age group classification and to calculate the loss, and then we calculate the accuracy achieved.

4.3.4 Gender Discriminator

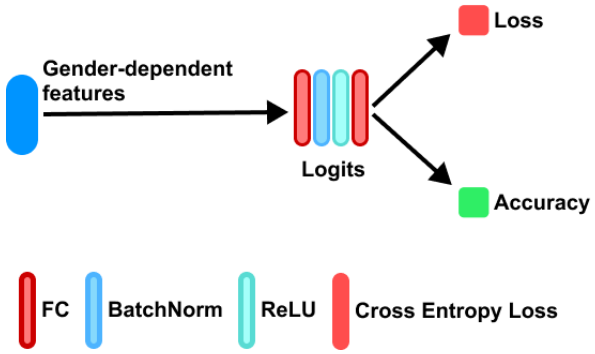

Figure 4.8: Gender Discriminator architecture.

To learn gender information, we feed x_{gender} into the Gender Discriminator to guarantee the gender-discriminating data. Our data contains gender labels and the classification of 2 genders, $male$ or $female$, is performed. We stack one Fully Connected Layer, Batch Normalization, ReLU activation, and another Fully Connected Layer over the x_{gender} features to calculate the logits, then use the Cross-Entropy Loss for the gender classification and to calculate the loss, and then we calculate the accuracy achieved.

4.3.5 Supervised Learning

Supervised learning in the multi-task framework entails the strategic training of a model using labeled data across several related tasks. In this case, the model is trained simultaneously on identity, age, and gender recognition tasks, leveraging a composite loss function that encompasses contributions from each specific task. This approach not only enhances the efficiency of the learning process but also improves the generalization capabilities of the model across diverse facial attributes.

The training is supervised by a unified loss function, defined as follows:

$$TL = L_{CE}(x_{id}) + \lambda_1 L_{CE}(x_{age}) + \lambda_2 L_{CE}(x_{gender}) \tag{4.6}$$

where TL denotes the total loss, comprised of the sum of the Cross-Entropy Loss L_{CE} evaluated on the identity features x_{id}, age features x_{age}, and gender features x_{gender}. The parameters λ_1 and λ_2 are weighting coefficients that adjust the impact of each respective loss component. During training, the model learns to navigate the complex landscape of facial feature differentiation. By minimizing the total loss TL, the model strives to improve its accuracy across all specified tasks, fostering a deep understanding of how different attributes interrelate. This is a key advantage resulting in a more robust model compared to the one trained for a single task.

4.4 Improvement Multi-Task Model with Decorrelated Adversial Learning

Facial attributes such as identity, age, and gender inherently comprise complex, intertwined features. Traditional facial analysis systems often address these attributes independently, which might overlook the intrinsic correlations among them. In the pursuit of an integrated approach that acknowledges and actively manages these correlations, we adopt the Multi-Task Model to a Decorrelated Adversarial Learning (DAL) method.

This innovative approach is influenced by prior studies, which have typically segregated these attributes without considering their potential interdependencies [HW19]. For instance, High-level Feature Attribution (HFA) [DGT13], Latent Feature Convolutional Neural Network (LF-CNN) [YWQ16], and Orthogonal Embedding CNN (OE-CNN) [YWZ18] have shown limitations in addressing the underlying connections between facial features. In response, our model adopts the Decorrelated Adversarial Learning method, initially proposed in [HW19], to minimize the mutual correlations between identity (x_{id}), age (x_{age}), and gender (x_{gender}) features.

The model's architecture, pictured in 4.9, facilitates simultaneous learning and refinement of each attribute's discriminative power while also enhancing the separation of these intertwined features. This dual-objective process is realized through a novel adversarial framework that decorrelates the features, ensuring that improvements in one attribute's accuracy do not detrimentally affect the others. This strategy not only improves the robustness of the model but also enhances its general applicability across diverse demographic groups.

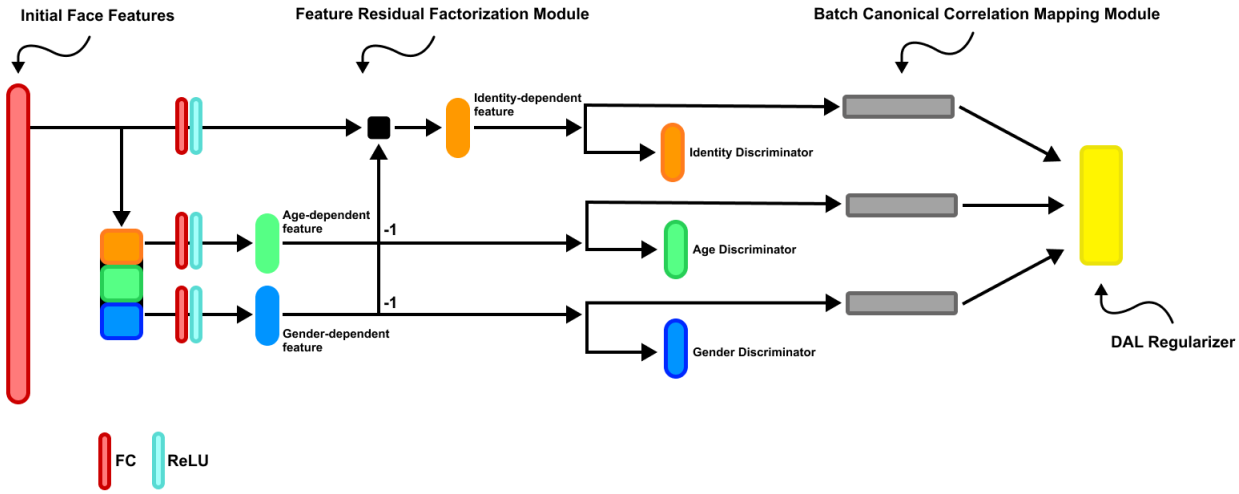

Figure 4.9: Overview on how the three factorized components x_{age}, x_{gender}, and x_{id} are used for classification and DAL regularization.

This section outlines the theoretical foundation and framework of the Decorrelated Adversarial Learning approach within our Multi-Task Model. By integrating this technique, the model not only achieves superior accuracy in attribute classification but also ensures that these attributes are processed in a mutually exclusive manner, thereby preserving the unique information each represents.

4.4.1 Batch Canonical Correlation Mapping Module

The Batch Canonical Correlation Mapping Module (BCCM), similar to [HW19], is an integral part of our Multi-Task Model, designed to quantify and maximize the correlations among the three decomposed facial features: identity (x_{id}), age (x_{age}), and gender (x_{gender}). This module leverages statistical techniques to capture and optimize inter-feature relationships, enhancing the model's ability to discriminate and correlate these attributes effectively.

The BCCM class is initialized with an embedding size, which dictates the dimensionality of the input feature vectors. Each feature set (identity, age, gender) is associated with a linear predictor that reduces the feature dimension to one. This reduction is essential for calculating scalar correlations between the different feature types.

During the forward pass, the method takes three inputs: identity features, age features, and gender features. Each set of features is first transformed using its corresponding predictor. The outputs, referred to as predictions, are then used to calculate the means and variances for each feature set. Following the computation of means and variances, pairwise correlations between each set of features are calculated using the formula for covariance divided by the product of the standard deviations of each feature set.

$$\text{id\_age\_corr} = \frac{(\text{age\_pred} - \text{age\_mean}) * (\text{id\_pred} - \text{id\_mean})}{\sqrt{\text{age\_var} * \text{id\_var}}}$$

$$\text{id\_gender\_corr} = \frac{(\text{gender\_pred} - \text{gender\_mean}) * (\text{id\_pred} - \text{id\_mean})}{\sqrt{\text{gender\_var} * \text{id\_var}}}$$

$$\text{age\_gender\_corr} = \frac{(\text{age\_pred} - \text{age\_mean}) * (\text{gender\_pred} - \text{gender\_mean})}{\sqrt{\text{age\_var} * \text{gender\_var}}}$$

Finally, the overall correlation coefficient is obtained by averaging these three pairwise correlation coefficients. This coefficient provides a scalar value that quantifies the overall interdependence among the identity, age, and gender features.

This module offers a powerful way for understanding and managing feature interdependencies in multidimensional data.

4.4.2 Decorrelated Adversarial Learning and Regularizer

We implement a Decorrelated Adversarial Learning (DAL) approach to fine-tune the process of strengthening both inter-subject and intra-subject correlations. This module is pivotal in managing how different facial attributes interact within the learning environment. Using DAL, the model dynamically adjusts to either maximize or minimize the correlation between the three feature sets based on the training objectives.

To effectively supervise the learning of decomposed facial features, our model employs a training strategy powered by four key supervision modules: the Identity Discriminator, Age Discriminator, Gender Discriminator, and the Decorrelated Adversarial Learning (DAL) Regularizer, as illustrated in Figure 4.9.

The DAL Regularizer plays an important role in training the model by guiding the feature learning process to maximize the correlations calculated by the BCCM between the decomposed features, namely x_{id}, x_{age}, and x_{gender}. This regularization technique, rooted in adversarial learning principles, aims to ensure that information extracted for each feature remains distinct and invariant, thus enhancing the model's ability to generalize across different tasks.

A key feature of the DAL Regularizer is the strategic manipulation of gradients during the training process, which is crucial for effectively minimizing or maximizing the correlation between features depending on the learning objectives. In the minimization process, we train only the Backbone and the FRFM parameters and freeze the BCCM parameters. In the maximization process, we freeze the Backbone and the FRFM and train the BCCM parameters, while also inversing its gradients for adversarial training and to ensure the maximization of the correlation between the decomposed features.

Following [HW19], the training of this model is governed by a combined multi-task loss function designed to minimize classification errors while simultaneously reducing correlation effects among the features:

$$TL = L_{CE}(x_{\text{id}}) + \lambda_1 L_{CE}(x_{\text{age}}) + \lambda_2 L_{CE}(x_{\text{gender}}) + \lambda_3 L_{DALR}(_{\text{id}}, x_{\text{age}}, x_{\text{gender}}) \quad (4.7)$$

where TL denotes the total loss, incorporating the Cross-Entropy Loss L_{CE} for identity, age, and gender features, alongside a specialized DAL Regularization Loss L_{DALR}. The coefficients λ_1, λ_2, and λ_3 are tunable hyperparameters that help balance the impact of each loss component on the overall training process.

These methods highlight the adversarial nature of the training process, which not only challenges the model to improve its discriminative abilities but also ensures that it does not overly specialize in correlated features, thereby enhancing its generalization capabilities across diverse datasets.

4.5 REST API

In order to have a scalable and accessible service that calculates the similarity between two facial images or two batches of images we combined web technologies and machine learning to create an easy-to-use REST API.

4.5.1 Requirements

The API provides two types of requests for our particular use case:

1. Comparing two photos to determine their identity similarity.
2. Comparing two batches of photos (with a minimum of one photo in the batch) to ascertain their identity similarities.

Requests have bodies that are form-encoded, and the API responds with JSON-encoded data. It employs conventional HTTP response codes and verbs to ensure clear communication of request outcomes.

4.5.2 System Design

The architecture consists of the following key components:

- **Configuration Management**: Implemented as a Singleton class to ensure a single source of truth for the configuration parameters.
- **Model Loading and Inference**: The deep learning model, Multi-Task + DAL, is loaded with pre-trained weights.
- **Similarity Handler**: Responsible for calculating the similarity between faces using the model.
- **Application**: Manages the API endpoints for handling requests.

The API offers specific endpoints for processing the image comparisons:

1. The `/images_similarity` endpoint handles POST requests for comparing two images.
2. The `/batch_images_similarity` endpoint handles POST requests for comparing two batches of images.

After confirming the request, the system first verifies the integrity and format of the images. Valid images then undergo a series of transformations to prepare them for analysis by the machine learning model. The first image transformations are:

- Resizing images to standardized dimensions (160x160 pixels).
- Cropping centrally to emphasize important facial features.
- Normalizing the images to standardize the inputs to the model, ensuring that the input data has zero mean and unit variance.
- Converting images into tensors to facilitate computations in the neural network.

Aside from these, other transformations are utilized:

- Horizontally flipping the images to augment the data, which improves the robustness of the similarity assessment.

The pre-trained model, tailored for extracting facial features or embeddings, processes the transformed images. Each image is loaded and transformed twice: once in its original form and once in its flipped form, using the predefined transformations from above to standardize and augment the input data. These images are fed into the neural network model, which computes the facial feature vectors (embeddings) without further adjustments due to the pre-trained state of the model. To enhance the feature set, the feature vectors of the original and flipped images are combined for each image. Cosine similarity is calculated between the combined feature sets of the two images, producing a similarity score. This score ranges from -1 (completely different) to 1 (identical), providing a quantitative measure of facial similarity. Finally, the service responds with the similarity score as a JSON response.

4.5.3 Implementation

The API is built using Flask, a lightweight web framework in Python, chosen for its simplicity and efficiency in creating web services. Flask serves as the backbone of our API, handling HTTP requests and responses.

We also use several key libraries and frameworks:

- **PyTorch:** A powerful deep learning library that is used for both loading the model and performing transformations and computations on the images.
- **Torchvision** A package from the PyTorch ecosystem that provides common transforms. We use it extensively for image transformations to prepare images for the model.
- **nn.Functional:** This module from PyTorch provides functions for calculating the cosine similarity between the combined feature vectors that were retrieved from the model.

- **PIL (Python Imaging Library):** This library is used to open, manipulate, and save many different image file formats. It is essential for initial image handling before processing.

REQUEST	TYPE	PARAMS		RESPONSE
/images_similarity	POST	KEY	VALUE	{ "similarity": -0.70 }
		image1	File img1.jpg	
		image2	File img1neg.jpg	can throw **invalid_request_error** or **request_error**
/batch_images_similarity	POST	KEY	VALUE	{ "similarity": 0.89 }
		imageList1	File 4 files	
		imageList2	File 4 files	can throw **invalid_request_error** or **request_error**

Figure 4.10: Requests available in the MPAIFR API, together with the type and response.

The endpoints pictured in 4.10, are the POST requests that, after confirming the validity of the request, examine the photos, process and augment the images with the above transformations from the Torchvision library, calculate the cosine similarity between the combined feature sets of both the image and its inverted version using nn.Functional, and respond with the similarity score as a JSON response.

HTTP STATUS	CODE	SUMMARY
OK	200	Everything worked as expected.
Bad Request	400	The request was unacceptable, due to missing a required parameter or invalid parameter type.
Request Failed	402	The parameters were valid but the request failed.

Figure 4.11: HTTP response codes available in the MPAIFR API.

As pictured in 4.11, we use standard HTTP response codes to signal the success or failure of an API call. Code 200 signifies success, while code 400 signifies an error indicating that the request failed because of the parameters provided. Code 402 also signifies a request error, but this one indicates that something failed on the server side.

ERROR	SUMMARY
invalid_request_error	Invalid request errors arise when the request has missing or invalid parameters.
request_error	Request errors arise when the request failed because of another type of problem (e.g. server is down)

Figure 4.12: Errors available in the MPAIFR API.

To make sure incoming requests contain two pictures, the API verifies them. It replies with the invalid_request error pictured in 4.12 and a 400 status code if the validation is unsuccessful or if a required parameter was omitted. If something fails on the server side, it replies with the request_error error also pictured in 4.12.

4.5.4 Deployment

Docker Containerization. The deployment leverages Docker to encapsulate the application and its dependencies into a single container. This approach ensures that the model, along with the necessary Flask application for serving model inference, can run consistently across various environments. The Docker container is built from a Docker image that includes all required libraries and the model itself, encapsulated to maintain the integrity of the development environment. This method simplifies the deployment process and ensures reproducibility and scalability.

Ngrok Delivery. Ngrok is used to create a secure tunnel to the local machine, allowing external access to the application running in the Docker container. The deployment process starts with the Ngrok agent establishing a connection to the Ngrok server. This server then sets up a tunnel that securely forwards requests from users to the Flask application inside the Docker container. This setup allows the application to be accessed over the internet without exposing the local network to potential security risks. Ngrok provides a public URL that routes traffic to the local server.

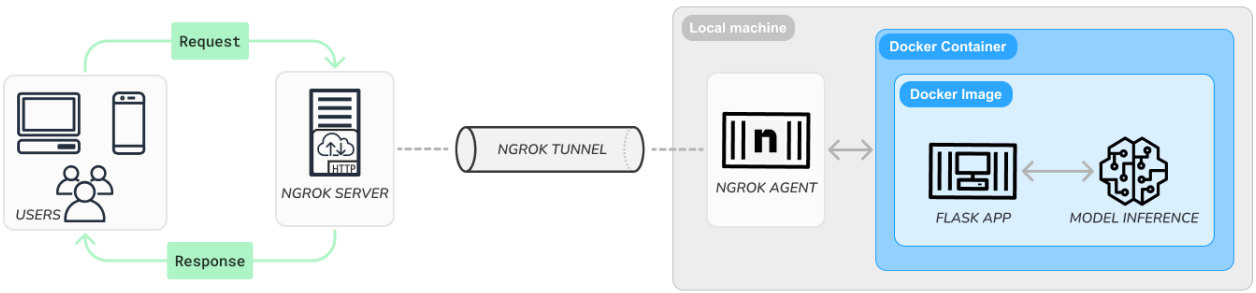

Figure 4.13: Deployment architecture.

4.6 Web Application

In this chapter, we describe the development of our Age-Invariant Face Recognition use case web application using Ruby on Rails. The app facilitates managing and verifying missing person profiles, leveraging our advanced Multi-Task+DAL model to identify potential matches.

4.6.1 Use Case Specification and Requirements Elicitation

Through the use of Age-Invariant Face Recognition, the application compares uploaded images with pre-existing profiles, thereby offering law enforcement and concerned parties an essential tool for the prompt identification and location of missing individuals. This system is a useful tool in the continuous efforts to reunite families and guarantee people's safety because it not only makes managing missing person cases more efficient, but it also improves communication between the public and authorities.

Regular Users: use the application to search for missing persons.

Admin Users: manage the application, profiles, and inquiries.

Developers: are responsible for developing and maintaining the application.

Customers: are entities or organizations that fund or utilize the application, such as law enforcement agencies and NGOs.

User Registration and Login

Users (both regular and admin) must be able to register and log into the application using a valid email address. Essential functionality for user access and security.

Dashboard Access for Regular Users

Regular users should have access to a dashboard displaying profiles consisting of images and details such as gender, nationality, and age. Core feature for user engagement and utility.

Inquiry Submission by Regular Users

Regular users should be able to send inquiries with images of a person that is a potential match for a profile, including additional information (e.g., date, city, country where the image was taken). Critical for the application's primary function of identifying missing persons.

Enhanced Dashboard for Admin Users

Admin users should have access to an enhanced dashboard with detailed profiles, including name, country, city of disappearance, and date of disappearance. Admins should be able to manage all profiles. This provides comprehensive management tools for admin users.

Inquiry Submission and Management by Admin Users

Admin users should be able to send inquiries with images of a person that is a potential match for a profile, including additional information (e.g., date, city, country where the image was taken). They should also view and manage all inquiries submitted by users. Essential for the administrative oversight and processing of inquiries.

Inquiry Review and Labeling

If an image from a submission matches an existing profile picture, the inquiry should be labeled as Found, Not Found, or Unverified based on the similarity score. Still admin are allowed to change the label of the match. Important for validating and managing the outcomes of inquiries.

User Management by Admin Users

Admin users must be able to register and log into the application and manage other users and roles. Necessary for the administration and maintenance of the application.

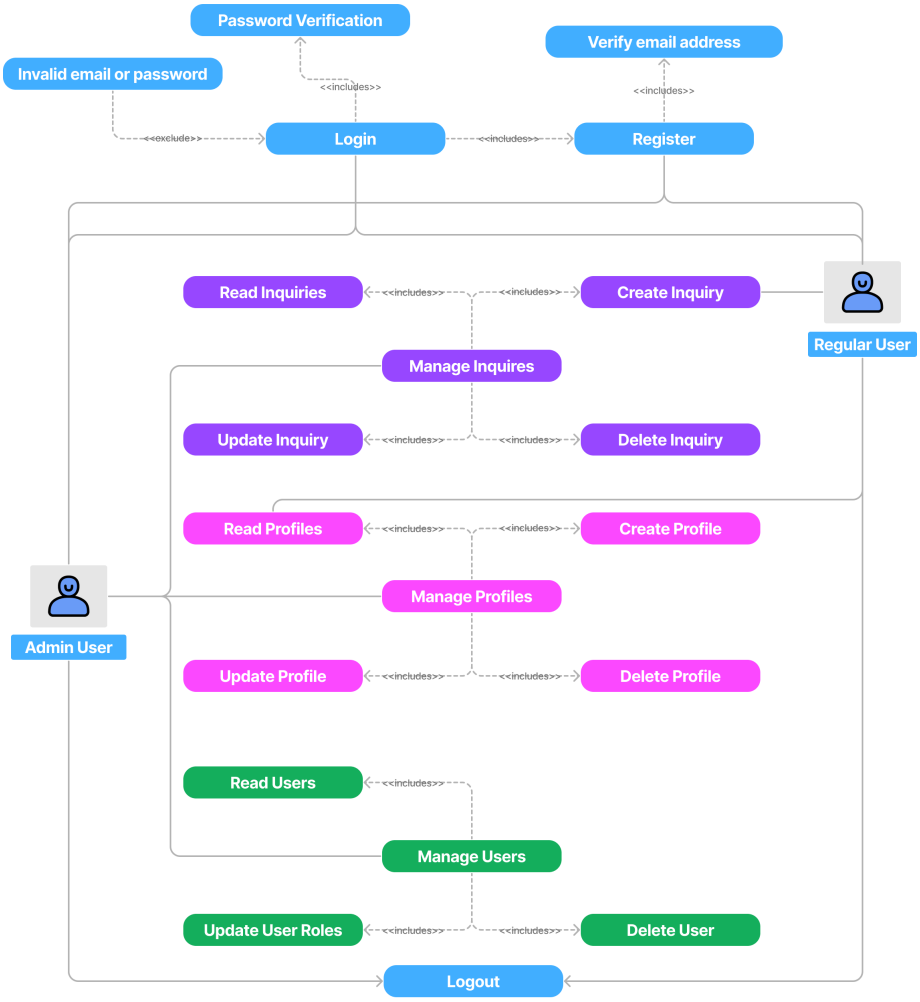

Figure 4.14: Use case diagram of the application.

4.6.2 System Design and Implementation

The application is structured around the Model-View-Controller (MVC) architecture, ensuring a clean separation of concerns and maintainable code. This design pattern is fundamental in Rails applications, dividing the application logic into three interconnected components: the model, view, and controller.

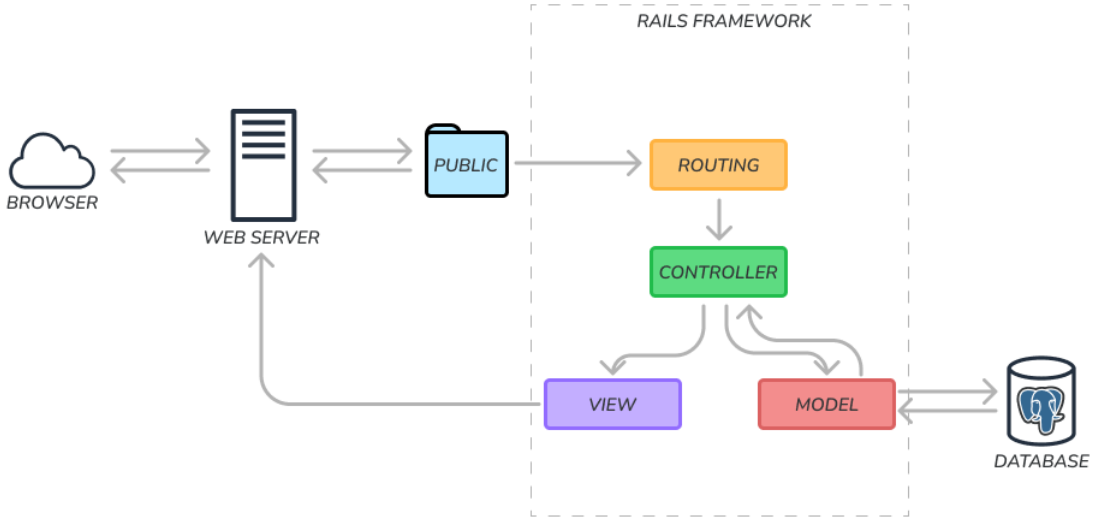

Figure 4.15: Application architecture.

Controllers manage the business logic for all models, including actions such as create, update, or delete. The controllers are the central hub where user requests are received, processed, and the appropriate responses are determined. The application fetches similarity scores for images using the external API via HTTP requests. The Faraday gem is used to handle multipart file uploads and process the API responses.

Views render the user interfaces for the dashboards, profile pages, and inquiry forms. Rendered using ERB (Embedded Ruby) templates, they form the user interface of the application. These views ensure that users can interact with the application intuitively and effectively. Bootstrap is used for responsive and mobile-friendly design, ensuring a consistent and user-friendly experience across devices.

Devise handles user authentication, including registration, login, password recovery, and email confirmation. This gem streamlines the process of securing user data and managing sessions. Rolify manages role-based access control (RBAC), assigning default roles upon user creation and allowing for role-based access control. This ensures that admin users have elevated permissions while regular users have restricted access. Active Storage manages file uploads, including images associated with profiles and inquiries. This allows users to attach multiple images to profiles

and inquiries, which are essential for accurate identification and verification.

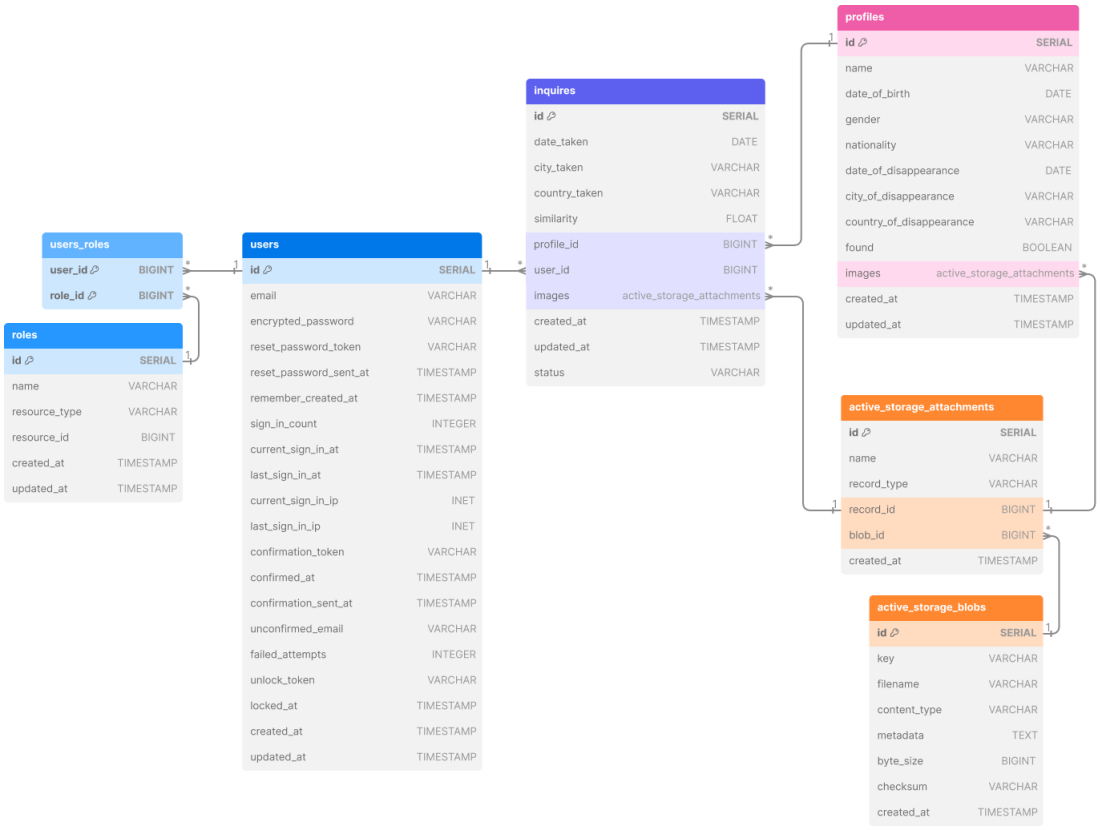

Figure 4.16: Database schema and tables relations.

As pictured in Figure 4.16, a PostgreSQL database stores user data, profiles, inquiries, and other relevant information. The database schema includes several key tables:

- **users**: Stores user information, including email, encrypted password, and authentication details. Devise is used to handle user authentication and sessions.
- **profiles**: Contains comprehensive details about missing persons, including their name, date of birth, gender, nationality, and details of their disappearance. Profiles are crucial for maintaining organized and searchable records.
- **inquiries**: Records details of images submitted by users, including the date, city, and country where the image was taken, along with the calculated similarity score and verification status. This table is essential for tracking the progress of inquiries and potential matches.
- **roles**: Defines different user roles within the application, such as regular users and admins. Managed by Rolify, this table helps enforce role-based access control.

- **users_roles**: A join table that establishes a many-to-many relationship between users and roles, ensuring that users can have multiple roles and roles can be assigned to multiple users.
- **active_storage_attachments** and **active_storage_blobs**: These tables store metadata and binary data of the uploaded files.

This comprehensive system design ensures that the application can efficiently manage user data, handle complex queries, and provide a seamless user experience.

4.6.3 Deployment

Our Ruby on Rails application is deployed using the cloud platform Heroku, which guarantees a stable, scalable, and user-friendly deployment environment. The application has been set up with the required environment variables and dependencies for Heroku deployment. Through Heroku's setup settings, environment variables like the database URL and secret keys are safely maintained.

The database is PostgreSQL, and Heroku's managed PostgreSQL service offers high availability, scalability, and automatic backups. During the deployment process, database migrations are automatically carried out to make sure the database schema is current.

Establishing a connection between the Heroku app and a GitHub repository allows continuous deployment. Heroku launches an automated build and updates the application whenever changes are pushed to the main branch. This configuration minimizes downtime and manual work by guaranteeing that the most recent features and fixes are always available.

The application runs on a Heroku dyno after it is launched, which is a lightweight container. These dynos house the Rails server and any background tasks that may exist. Heroku controls scaling by permitting the deployment of more dynos in response to traffic needs, guaranteeing that the application can accommodate increased load.

Heroku runs automated tests (if configured) and deploys the application to a staging environment for final verification. Once the build passes all tests, it is deployed to the production environment. Heroku ensures zero downtime by managing rolling deployments.

Strong monitoring tools are available from Heroku to keep an eye on logs, resource utilization, and application performance.

By using Heroku for deployment, we can take advantage of a robust platform that abstracts a large portion of the operational complexity, freeing us up to concentrate on creating and refining the application, and the web application can be delivered more effectively for usage.

4.6.4 Application Flows

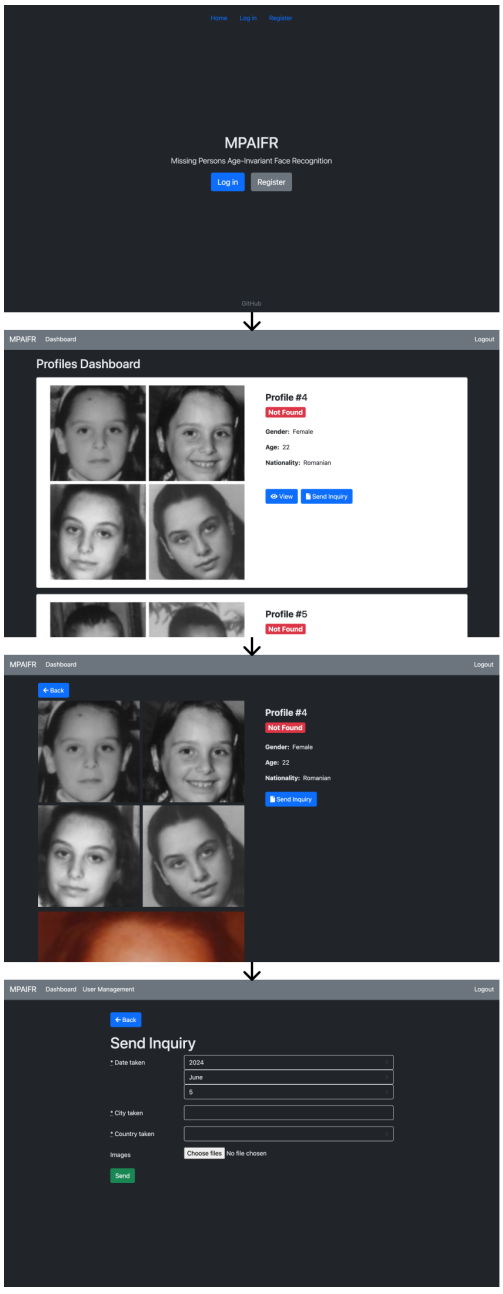

Figure 4.17: Regular user flow for sending an Inquiry to a Profile.

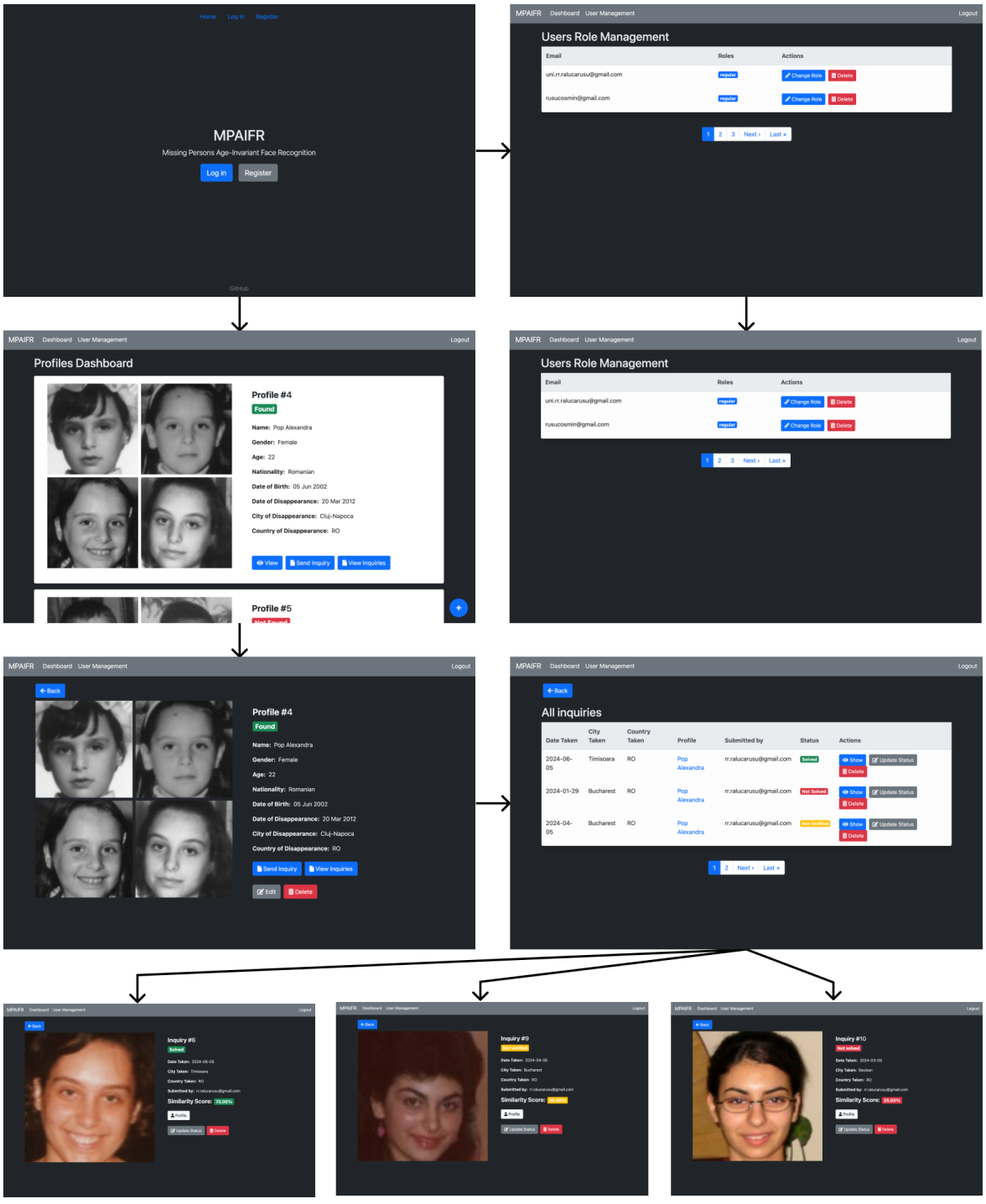

Figure 4.18: Admin user flows for managing Profiles, Inquires, and Users.

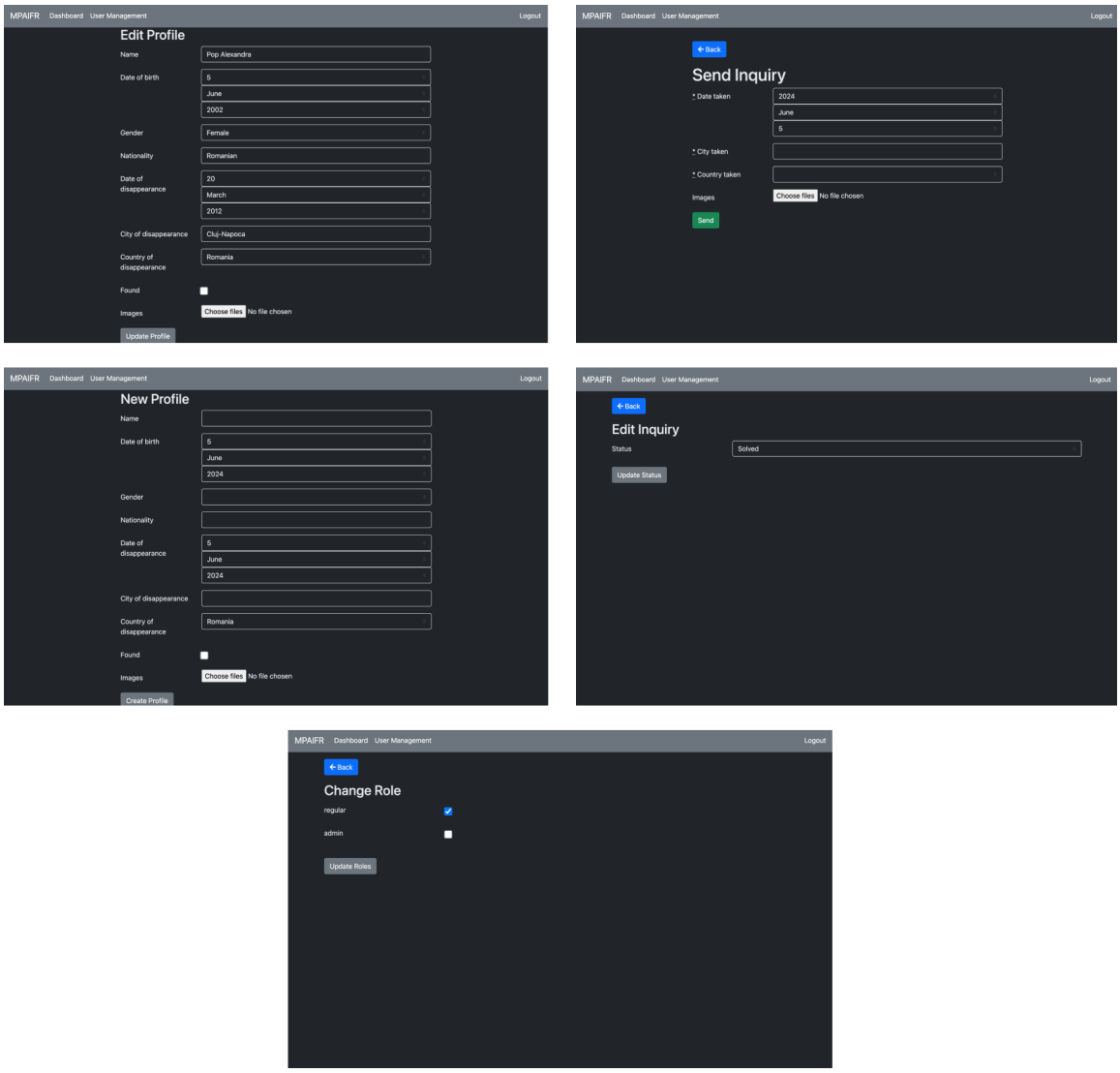

Figure 4.19: Other admin user operations over Profiles, Inquires, and Users.

Chapter 5

Experiments

5.1 Implementation Details

5.1.1 Datasets

In order to obtain the best results we needed a vast amount of qualitative data to feed the models. We used the AgeDB dataset [Mos17] and only a small part of the CACD dataset [Che15] to create our datasets. We performed manual verifications, additions, and changes for the actuality of the age and gender labels, the uniqueness of the identities, and the quality of the images. With the collected images we created two datasets in order to perform objective comparative tests.

Small dataset. Our small dataset contains 500 unique identities, with more than 20 images per identity, with an age range form 0 to 101. It has a total of 6626 images of female and 9126 images of male subjects. The total number of images is 15.752.

Large dataset. Our large dataset contains 1035 unique identities, with more than 5 images per identity, with an age range form 0 to 101. It has a total of 8244 number of females and 12143 number of males subjects. The total number of images is 20.387.

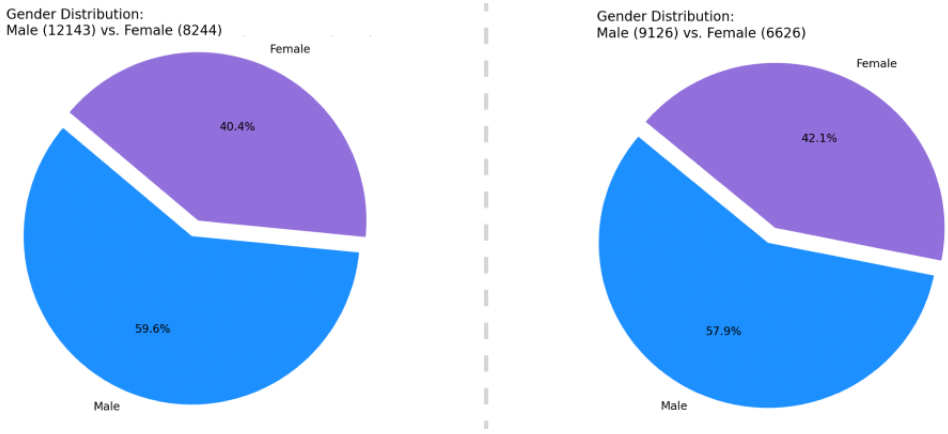

Figure 5.1: Large and small datasets gender distribution.

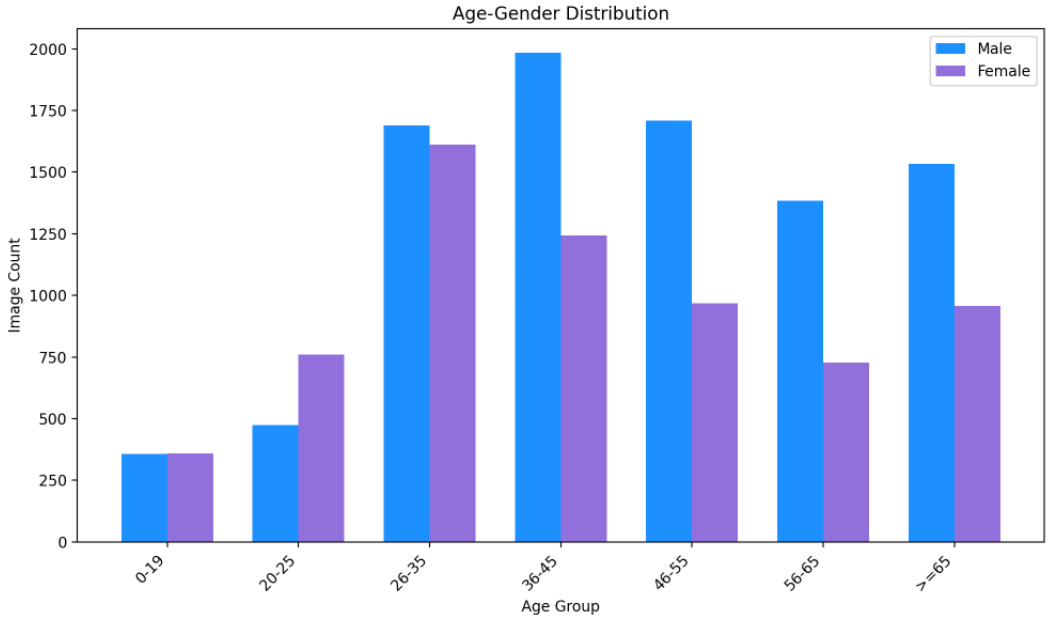

Figure 5.2: Small dataset age distribution by gender.

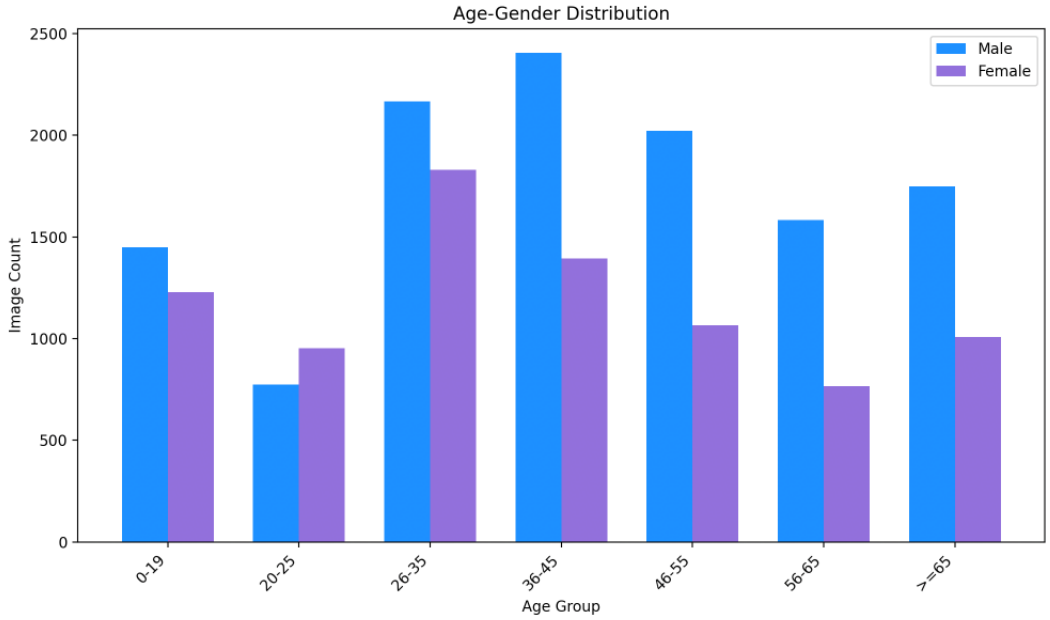

Figure 5.3: Large dataset age distribution by gender.

5.1.2 Data Processing

Developing reliable models requires proper data preprocessing, especially in the area of Face Recognition. The purpose of these adjustments is to improve the predicted accuracy and adaptability of the model by enhancing its capacity to generalize from the training data to the completely new data used in the evaluation.

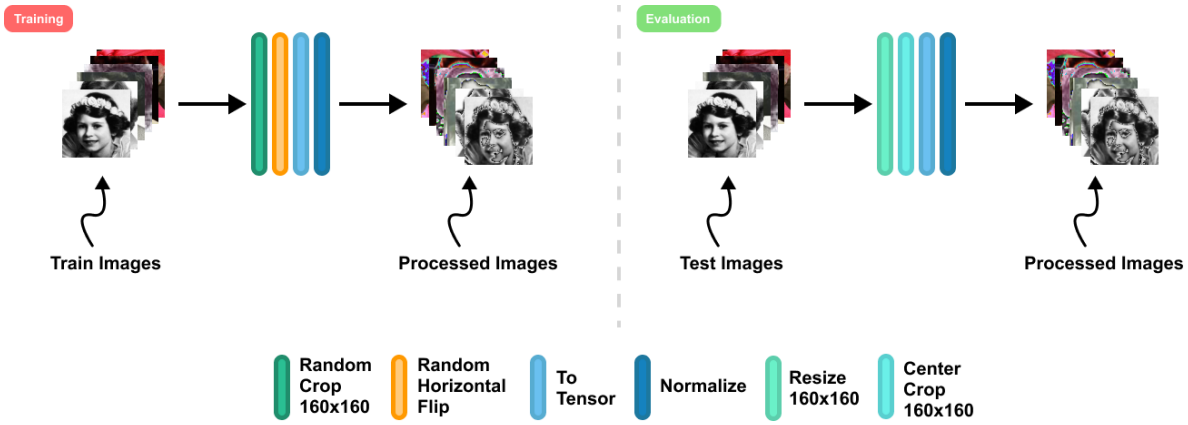

Figure 5.4: Data processing steps in both the training and evaluation phases.

In the training process, we used several transformations to improve the data and to simulate the variability that the model would face in real-world scenarios:

- Random resize cropping of images to a resolution of 160x160 pixels.
- Horizontally flipping the images with a 50% probability to augment the data.

In the evaluation process, we used a distinct set of transformations, in order to assess the model's performance in a controlled environment that replicates the variety of the training phase without adding more unpredictability.:

- Resize of images to ensure a uniform dimension of 160x160 pixels.
- Center cropping the images to maintain consistency in the input data size and focus.

In both processes, we also add to the list of transformations:

- Normalizing the pixel values of the images using predefined mean and standard deviation values ([0.485, 0.456, 0.406] for the mean and [0.229, 0.224, 0.225] for the standard deviation, respectively). This step is vital for standardizing the inputs to the model, enhancing training stability and convergence speed by ensuring that the input data has zero mean and unit variance.
- Converting images into tensors making them suitable for processing by the neural network.

The selection of image transformations maintains an equilibrium between presenting images to the model reliably, but also adding the required variety. The purpose of the training transformations is to increase the model's capacity for generalization by exposing it to a large variety of potential situations. On the other hand, the evaluation transformations are intended to assess how well the model performs.

5.1.3 Training Details

We trained the models on both the small and large datasets. Different datasets (big, small) and respective paths (train, test) are defined to diversify the training scenarios and ensure the model is robust across various facial recognition challenges. Custom data loaders are implemented for loading and batching the training and testing datasets, with specific transformations applied to augment the data and prevent overfitting. These transformations are defined in 4.1.2.

The Single-Task Model focuses on optimizing facial feature recognition with respect to the identity attribute, using a structured and targeted approach. This training regimen is fine-tuned through the configuration of several critical hyperparameters and operational settings, ensuring optimal learning outcomes.

The Multi-Task Model is designed to concurrently learn several facial attributes, enhancing the efficiency and effectiveness of the training process. This model leverages a configuration that integrates multiple discriminative tasks—identity, age, and gender recognition—into a unified training framework.

The Multi-Task Model with DAL integrates multiple facial attribute recognitions—identity, age, and gender—with an added layer of complexity involving adversarial learning to both minimize and maximize correlations among these attributes. This advanced model aims to enhance generalization by ensuring that the learned features for each attribute are as independent as possible.

Training is initialized through a configuration singleton class that sets key parameters and paths for the model's training, evaluation, and saving procedures.

- **Model Name and Weights.** The model is designated as 'singletask', 'multitask', or 'multitask_dal', with pre-trained weights loaded if available, facilitating a more focused and effective training phase, particularly for complex datasets.

- **Embedding size.** Set at 512, the embedding size parameter defines the dimensionality of the facial embeddings, providing sufficient granularity for capturing distinct facial features.

- **Loss Function.** The 'cosface' margin loss is employed, which introduces an angular margin to enhance feature discrimination.

- **Hyperparameters.** Hyperparameters such as the learning rate, batch size, and number of epochs (0.01, 64, and 40, respectively for all models) are configured to balance the speed and accuracy of convergence.

The model is instantiated through a handler that retrieves the appropriate model type based on the configuration. This model is then used to initialize a trainer class, which manages the training process. The trainer setup involves:

- **Optimizer:** An SGD optimizer with momentum (0.9) is employed to adjust model weights effectively, balancing the quick convergence and stability of the training process. In the case of the Single-Task model, it optimizes the parameters of the Backbone. In the case of the Multi-Task model, it optimizes the parameters of the Backbone and FRFM module. In the case of the Multi-Task + DAL model, it optimizes the parameters of the Backbone, FRFM module, and BCCM module, in a decorrelated adversarial manner.

Training execution for the Multi-Task + DAL model is characterized by its cyclic adjustment of learning modes to alternately focus on direct task learning and adversarial decorrelation:

- **Dynamic Training Mode Adjustment:** Training modes are dynamically adjusted to switch between focusing on maximizing correlation (using the BCCM) and standard multitask learning, which allows the model to develop a robust understanding of each attribute while maintaining their independence.

- **Adversarial Strategy:** For 40 iterations the model runs its minimizing procedure by optimizing the Backbone's and the FRFM module's parameters, while for other 30 iterations, it runs its maximizing procedure by flipping the BCCM module's gradients and optimizing only its parameters.

Training is executed through a loop where each epoch processes the input data through the model, computes losses, and updates weights:

- **Loss Computation and Backpropagation:** In the case of the Single-Task model, the loss is represented only by the identity loss, and backpropagation is conducted to adjust the model weights. In the case of the Multi-Task model, the total loss is calculated as a combination of identity, age, and gender losses, with each component weighted by predefined lambdas (1, 0.9, 0.9), and backpropagation is conducted to adjust the model weights. This approach ensures that all tasks contribute appropriately to the overall learning objective. In the case of the Multi-Task + DAL model, the total loss is calculated as a combination of identity, age, gender, and DALR losses, with each component weighted by predefined lambdas (1, 0.9, 0.9, 0.9), and backpropagation is conducted to adjust the model weights.

Training progress and key metrics are logged using Wandb, providing real-time insights into the model's performance and facilitating quick adjustments as needed.

At the end of each training epoch, the model's state is saved, facilitating periodic evaluations and long-term retention of the best-performing models. This approach ensures that the most effective version of the model is retained and available for further testing or operational deployment.

This structured training approach ensures that the Single-Task Model is not only optimized for high accuracy in facial feature recognition but also adaptable to different datasets and capable of robust performance in diverse deployment scenarios.

By integrating multiple facial recognition tasks into a single model, the Multi-Task Training approach not only enhances computational efficiency but also improves the model's ability to generalize across tasks. This training strategy ensures that the model is not only optimized for accuracy but also robust against overfitting specific facial attributes.

The integration of DAL into the Multi-Task Model training regime represents a significant enhancement in the model's ability to handle complex, interrelated tasks by learning to distinguish and independently modify facial attributes. This not only improves the model's accuracy but also its robustness and adaptability in diverse application scenarios.

5.1.4 Testing Details

We conduct evaluation experiments on the well-known public AIFR face datasets FG-NET [Lan02] and 20% of our created datasets (not used in training). In the testing process, we extract the identity-dependent features and concatenate features of the original image and the flipped image to form the final representation. The cosine similarity of these representations is then used to conduct face verification and identification.

The evaluations detailed in 4.2 highlight the model's capabilities in not only distinguishing individuals based on their facial features but also tracking these features consistently across different images, potentially taken under varying conditions and at different times. The accuracy and reliability of these tests are crucial for our use case application of matching images of missing persons, where precise and robust facial recognition is necessary.

5.2 Evaluation

5.2.1 Models Comparative Analysis

The evaluation of the Single-Task, Multi-Task, and Multi-Task + DAL models was conducted using an 80/20 training/testing data split. This common evaluation methodology allows for a balanced comparison across models, ensuring that each model is tested under identical conditions to fairly assess its performance capabilities.

80/20 Split. The models were evaluated on both large and small datasets to further understand their scalability and efficiency in handling datasets of different sizes. The performance of each model is reported in terms of accuracy, which is a critical metric for facial recognition tasks.

Model	**Trained on**	**80/20 Split**
Single-Task	large	85.65%
Multi-Task	large	85.92%
Multi-Task + DAL	large	**86.24%**
Single-Task	small	93.02%
Multi-Task	small	93.12%
Multi-Task + DAL	small	**93.89%**

Table 5.1: Accuracy results for Single-Task, Multi-Task, and Multi-Task + DAL models on large and small datasets under the 80/20 split protocol.

From the results presented in Table 5.1, several observations can be made. The Multi-Task and Multi-Task + DAL models consistently outperform the Single-Task model across both dataset sizes. This suggests that integrating multiple learning tasks into a single model does not compromise, but can enhance the ability to effectively recognize facial features. The Multi-Task + DAL model shows a slight improvement over the standard Multi-Task model, particularly in the large dataset scenario. This improvement underscores the benefit of using DAL to minimize feature correlation, which potentially enhances the model's ability to generalize across more complex or diverse datasets. The highest accuracy is observed in the Multi-Task + DAL model trained on the small dataset, which may indicate that DAL's decorrelation capabilities are particularly effective when handling more nuanced data.

These findings suggest that the integration of multitasking learning frameworks, especially when combined with advanced techniques like DAL, can significantly improve the performance of facial recognition systems. The enhanced ability to handle complex interactions between different facial attributes in the Multi-Task +

DAL model also suggests promising applications in fields requiring robust, scalable, and highly accurate facial recognition capabilities.

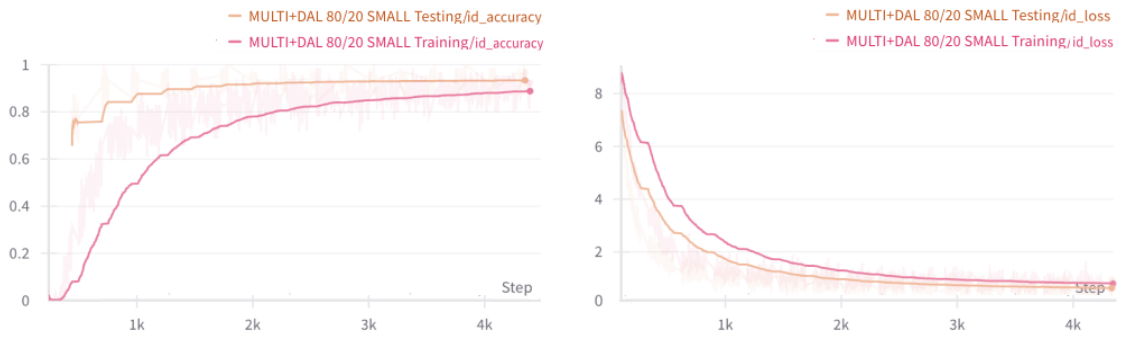

Figure 5.5: Wandb graphs of Multi-Task+DAL model training on the small dataset.

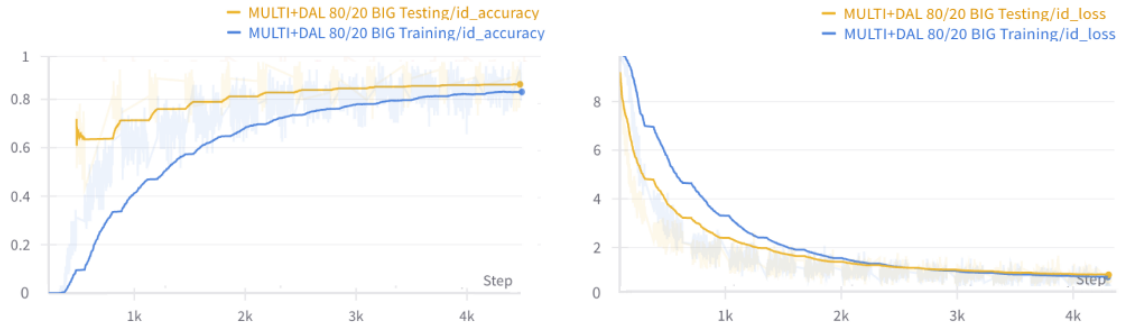

Figure 5.6: Wandb graphs of Multi-Task+DAL model training on the large dataset.

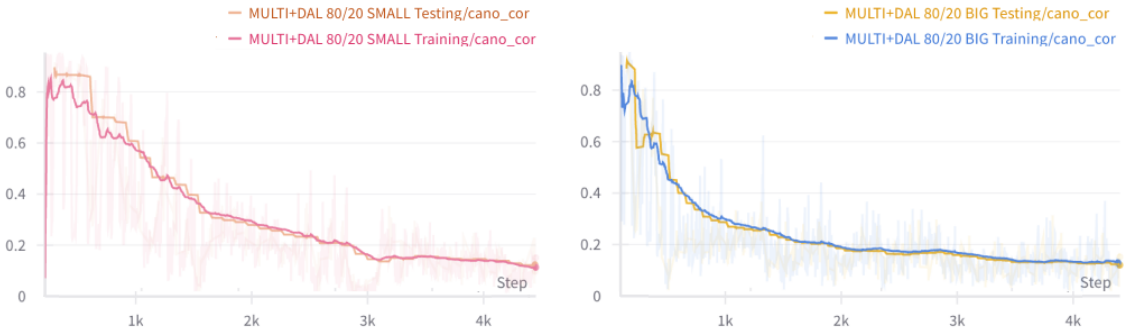

Figure 5.7: Wandb graphs of the canonical correlation minimizing and maximizing procedure while training on both datasets.

5.2.2 Experiments on the FG-NET Dataset

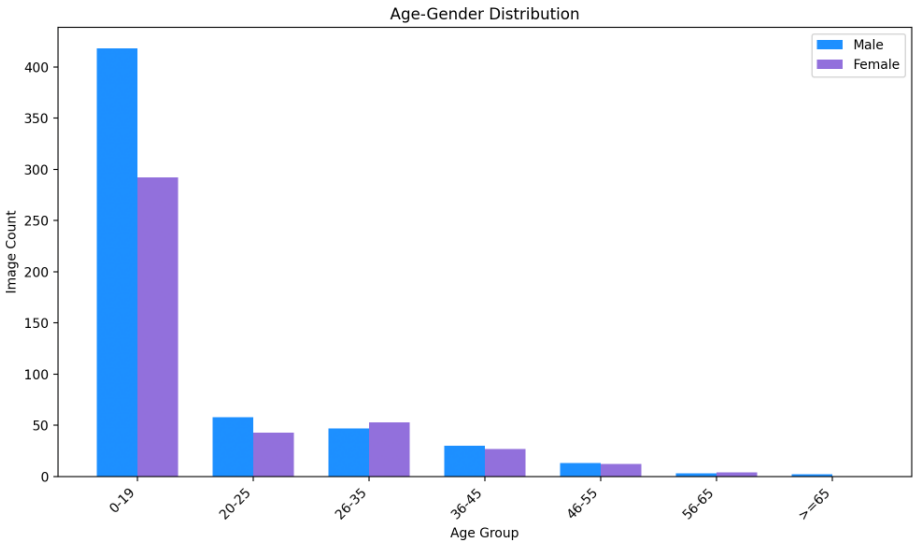

Figure 5.8: FG-NET dataset age distribution by gender.

The FG-NET Aging Dataset from [Lan02], extensively detailed in 2.3.1, consists of 1002 facial images representing 82 distinct individuals across various age groups. The age distribution, particularly prominent in the younger age brackets as illustrated in Figure 5.8, provides a robust foundation for evaluating age progression and regression models. The significant representation of the [0-19] and [0-25] age groups highlights the dataset's suitability for assessing model performance across wide age variances, which is critical for applications in Age-Invariant Face Recognition.

Two-Pairs Evaluation. This evaluation involved creating 1000 positive and 1000 negative pairs of images, with an equal distribution across genders and a stipulated minimum age difference of 15 years between the paired images. This method tests the model's ability to discern and verify facial features across significant age gaps, thus simulating real-world scenarios where age-related changes are pronounced.

For each pair, the model processes the transformed images to extract features. These features are then used to calculate the cosine similarity, a metric that quantifies the likeness between two feature sets.

By aggregating the features from both the original and flipped images, the evaluation aims to capture a more comprehensive representation of each face, enhancing the reliability of the similarity measure.

A predefined threshold of 0.5 is used to decide whether a pair of images likely represents the same person based on their similarity score. If the similarity score exceeds the threshold in a positive pair scenario, it is counted as a correct prediction. Conversely, if the score falls below the threshold for a negative pair, this also

constitutes a correct prediction.

The outcomes of these comparisons are then aggregated to calculate the overall accuracy of the model in distinguishing between positive and negative scenarios.

Model	**Trained on**	**Evaluation two-pairs**
Single-Task	large	79.31%
Multi-Task	large	79.83%
Multi-Task + DAL	large	**80.00%**
Single-Task	small	78.85%
Multi-Task	small	79.42%
Multi-Task + DAL	small	**82.21%**

Table 5.2: Performance comparison of models on two-pair evaluation using the FG-NET dataset.

We tested all three models, Single-Task, Multi-Task, and Multi-Task + DAL, trained on both small and large training datasets. The evaluation results, as summarized in Table 5.2, demonstrate that the Multi-Task + DAL model consistently outperformed the other models, underscoring the effectiveness of integrating multi-task and decorrelated adversarial learning in improving recognition accuracy across age-diverse facial data.

Leave-one-out Evaluation. In this rigorous testing protocol, each image in the FG-NET dataset was used once as a test image while the remainder formed the training set. This process was iterated until all images had been used as the test image. Importantly, no images from FG-NET were employed for training or fine-tuning the models prior to this evaluation, ensuring that the test conditions were strictly unbiased and indicative of true model performance in unseen scenarios.

Method	**Rank-1**
Park et al. (2010) [UP10]	37.40%
Li et al. (2011) [ZL11]	47.50%
HFA (2013) [DG13]	69.00%
MEFA (2015) [DG15]	76.20%
CAN (2017) [CX17]	86.50%
LFCNNs (2017) [CND17]	88.10%
AIM (2018) [JZ18]	93.20%
Age + DAL (2019) [HW19]	94.50%
Multi-Task + DAL	**94.61%**

Table 5.3: Comparative performance of our Multi-Task + DAL model against other methods under the leave-one-out evaluation protocol on the FG-NET dataset.

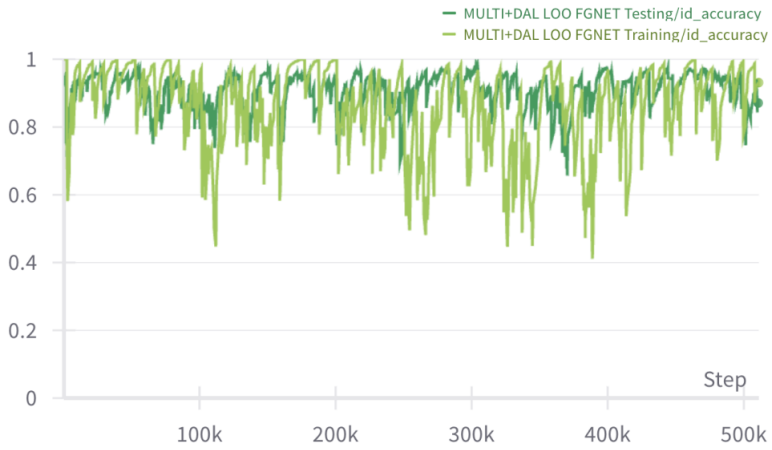

Figure 5.9: Wandb graph of Multi-Task+DAL model LOO evaluation.

The leave-one-out results, along with a comparative analysis with related works in facial recognition across age variations, are presented in Table 5.3. Our best-performing model, Multi-Task + DAL, achieved notable improvements over previous methods, highlighting advances enabled by our model architecture and training approach.

These evaluations rigorously confirm the capability of our Multi-Task + DAL model, to handle the challenges of age-diverse facial recognition, providing a strong foundation for further development and deployment in practical applications.

Chapter 6

Conclusions

This thesis has presented an innovative approach to Age-Invariant Face Recognition through the development and application of the Multi-Task+DAL model. The proposed methodology has shown significant improvements in recognizing faces across varying age groups, thereby addressing a critical challenge in the domain of Face Recognition. The findings from this research hold substantial implications for the enhancement of robust face recognition systems, particularly in the context of identifying and locating missing persons. The improved accuracy and reliability of the AIFR model have the potential to significantly enhance search and identification efforts, facilitating the reunification of families and contributing to public safety.

The comprehensive evaluation of the AIFR model has underscored its effectiveness and potential utility in real-world applications. By leveraging advanced techniques and incorporating a multi-task learning framework, this research has contributed to the field by offering a scalable and efficient solution for Age-Invariant Face Recognition.

Additionally, this thesis has successfully demonstrated the practical applicability of the AIFR model through its integration into an API. The development of the API showcases how the model can be utilized across various platforms and use cases, providing a flexible interface for developers to implement Age-Invariant Face Recognition capabilities within their applications.

Furthermore, the thesis has explored the implementation of a web application designed to aid in the search for missing persons. This application leverages the AIFR model to allow users to access profiles of missing individuals and to submit inquiries accompanied by images of potential matches. Admin users can review these inquiries, view the calculated similarity scores, and label the profiles accordingly if any match is found. This functionality provides a structured and efficient process for identifying potential matches, thus aiding in the resolution of missing-person cases. The web application not only demonstrates the practical benefits of the AIFR model but also highlights the potential for significant real-world impact.

By providing a tool that enhances the ability of law enforcement agencies to locate missing persons, the application underscores the social relevance and utility of the research conducted.

While acknowledging the limitations discovered, this thesis has made notable contributions to the field of FR by addressing the challenges associated with AIFR and by developing practical tools that can be used in critical applications. The AIFR model, API, and web application collectively offer a comprehensive solution that can be further refined and expanded to maximize their impact on public safety and the efficacy of search efforts for missing persons.

6.1 Future improvements

In the course of this research, several potential areas for future enhancement of the core development components—namely, the AIFR model, the API, and the web application—have been identified. Addressing these areas could lead to the development of an even more robust and scalable system.

For the AIFR model, future work could focus on increasing the accuracy of similarity scores by integrating more diverse and extensive datasets, as well as fine-tuning hyperparameters. Additionally, exploring the incorporation of real-time learning capabilities, wherein the model continuously learns from its operational environment, could further enhance its accuracy and effectiveness under varying conditions. Integrating additional biometric attributes, such as racial features, could also improve the overall accuracy and reliability of the recognition system.

Enhancements to the API should aim at expanding its functionality and optimizing its performance. This includes improving response times through better load balancing and caching strategies, as well as deploying the server on a cloud provider to ensure scalability and availability.

For the web application, potential improvements include the integration of more intuitive and user-friendly interface designs, based on iterative user feedback, to enhance user experience. Additionally, implementing new features that respond to user needs and requirements can increase the application's effectiveness. Considering a transition to a cloud-based file storage solution for managing images could also provide scalability and improve access speeds.

By focusing on these future improvements, this research can pave the way for the development of a more resilient and comprehensive Age-Invariant Face Recognition system. Such advancements will not only enhance the technical robustness of the solution but also expand its applicability and impact in real-world scenarios, particularly in aiding law enforcement agencies in locating missing persons.

Bibliography

[Che15] Chen Chu-Song Hsu Winston Chen, Bor-Chun. Face recognition and retrieval using cross-age reference coding with crossage celebrity dataset. *IEEE Transactions on Multimedia*, page 804–815, 2015.

[CND17] K. Luu M. Savvides C. N. Duong, K. G. Quach. Temporal non-volume preserving approach to facial ageprogression and age-invariant face recognition. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[CS16] V. Vanhoucke A. Alemi C. Szegedy, S. Ioffe. Inception-v4, inception-resnet and the impact of residual connections on learning, 2016.

[CX17] M. Ye C. Xu, Q. Liu. Age invariant face recognition and retrieval by coupled auto-encoder networks. *Neurocomputing*, 2017.

[DG13] D. Lin J. Liu X. Tang D. Gong, Z. Li. Hidden factor analysis for age invariant face recognition. *International Conference on Computer Vision (ICCV)*, 2013.

[DG15] D. Tao J. Liu X. Li. D. Gong, Z. Li. A maximum entropy feature descriptor for age invariant face recognition. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, page 5289–5297, 2015.

[DGT13] D. Lin J. Liu D. Gong, Z. Li and X. Tang. Hidden factor analysis for age invariant face recognition, 2013.

[Esl17] Tim Esler. Face recognition using pytorch, 2017.

[GZ19] G. Guo and N. Zhang. A survey on deep learning based face recognition. *Computer Vision and Image Understanding*, 189:102805, 2019.

[Hua23] Zhang Junping Shan Hongming Huang, Zhizhong. When age-invariant face recognition meets face age synthesis: A multi-task learning framework and a new benchmark. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45:7917–7932, 2023.

[HW18] Z. Zhou X. Ji D. Gong J. Zhou Z. Li W. Liu H. Wang, Y. Wang. Cosface: Large margin cosine loss for deep face recognition. *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[HW19] Zhifeng Li Wei Liu Hao Wang, Dihong Gong. Decorrelated adversarial learning for age-invariant face recognition, 2019.

[Isl21] Lee Sujin Han Dongil Moon Hyeonjoon Islam, Khawar. Face recognition using shallow age-invariant data. pages 1–6, 2021.

[JW06] G. Su X. Lin J. Wang, Y. Shang. *Age simulation for face recognition*, volume 3. 2006.

[JZ18] Yi Cheng Y. Yang H. Lan F. Zhao L. Xiong Y. Xu J. Li S. Pranata S. Shen J. Xing H. Liu S. Yan J. Feng J. Zhao, Yu Cheng. Look across elapse: Disentangled representation learning and photorealistic cross-age face synthesis for age-invariant face recognition, 2018.

[KC18] C. Li X. Tang C. C. Loy K. Cao, Y. Rong. Pose-robust face recognition via deep residual equivariant mapping. *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5187–5196, 2018.

[Lan02] Taylor C.J. Cootes T.F. Lanitis, A. Toward automatic simulation of aging effects on face images. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2002.

[MMS19] K. M. Bhurchandi M. M. Sawant. Age invariant face recognition: a survey on facial aging databases, techniques and effect of aging. *Artificial Intelligence Review*, 52:1–28, 2019.

[Mos17] Papaioannou Athanasios Sagonas Christos Deng Jiankang Kotsia Irene Zafeiriou Stefanos Moschoglou, Stylianos. Agedb: The first manually collected, in-the-wild age database. pages 1997–2005, 2017.

[Nay22] Indiramma M. Nayak, J. S. An approach to enhance age invariant face recognition performance based on gender classification. *Journal of King Saud University - Computer and Information Sciences*, 34:5183–5191, 2022.

[Par08] Tong Yiying Jain Anil K. Park, Unsang. Face recognition with temporal invariance: A 3d aging model. In *2008 8th IEEE International Conference on Automatic Face Gesture Recognition*, pages 1–7, 2008.

[Par19] Tong Yiying Jain Anil Park, Unsang. Facial aging databases, techniques and effects of aging: A survey. *International Journal of Engineering Trends and Technology (IJETT)*, 67:8–13, 2019.

[QC18] W. Xie O. M. Parkhi A. Zisserman Q. Cao, L. Shen. Vggface2: A dataset for recognising faces across pose and age. In *International Conference on Automatic Face and Gesture Recognition*, 2018.

[Ric06] Tesafaye T. Ricanek, Karl. Morph: A longitudinal image database of normal adult age-progression. pages 341 – 345, 2006.

[SM23] S. Patel S. Mittal. Age invariant face recognition techniques: A survey on the recent developments, challenges, and potential future directions. *International Journal of Engineering Trends and Technology*, 71:1–26, 2023.

[UP10] A. K. Jain U. Park, Y. Tong. Age-invariant face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2010.

[Upr22] Anjeel Upreti. Convolutional neural network (cnn). a comprehensive overview. 08 2022.

[YWQ16] Z. Li Y. Wen and Y. Qiao. Latent factor guided convolutional neural networks for age-invariant face recognition, 2016.

[YWZ18] Z. Zhou X. Ji H. Wang Z. Li W. Liu Y. Wang, D.Gong and T. Zhang. Orthogonal deep features decomposition for age-invariant face recognition, 2018.

[ZL11] A. K. Jain Z. Li, U. Park. A discriminative model for age invariant face recognition. *IEEE transactions on Information Forensics and Security (TIFS)*, 2011.

List of abbreviations

AI Age-Invariant Face Recognition. 1

AIFR Age-Invariant Face Recognition. ii, 1–3, 6, 8–11, 13, 14, 21–45, 51, 57, 58

AIFV Age-Invariant Face Verification. 6, 10

API Application Programming Interface. iii, 2, 34–37, 40, 58

CNN Convolutional Neural Network. ii, 17–22, 28

DM Discriminative Methods. ii, 13

DNN Deep Neural Network. ii, 4, 8, 11, 13, 15

FR Face Recognition. ii, 1, 4, 5, 24, 58

GM Generative Methods. ii, 11

REST Representational State Transfer. iii, 2, 34

SOTA State-Of-The-Art. 14