



ESCUELA DE INGENIERÍA EN COMPUTACIÓN
PROGRAMA DE MAESTRÍA EN COMPUTACIÓN

A Deep learning algorithm to address kinship verification integrating age transformation techniques applied to the family images and model tuning methodologies

A thesis submitted in partial fulfillment to opt for the degree of

Magister Scientiæ in Computer Science

Author

Priscilla Piedra Hidalgo

Supervisor

Abel Méndez Porras

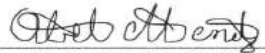
November 28, 2024

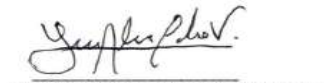
ACTA DE APROBACION DE TESIS

A Deep learning algorithm to address kinship verification integrating age transformation techniques applied to the family images and model tuning methodologies

Por: PRISCILLA MARIA PIEDRA HIDALGO

TRIBUNAL EXAMINADOR


Dr. Abel Méndez Porras
Profesor Asesor


Dr. Luis Alexander Calvo Valverde
Profesor Lector


Dr. Sixto Campaña Bastidas
Lector Externo


Dra.-Ing. Lilliana Sancho Chavarria
Presidente, Tribunal Evaluador Tesis
Programa Maestría en Computación



31 de octubre, 2024

To myself.

Acknowledgements

I would like to express my gratitude to my thesis advisor, Dr. Abel Méndez Porras, whose guidance and expertise were invaluable in the completion of this thesis.

Abstract

This thesis addresses the challenge of verifying familial relationships through facial features, which is often complicated by age-related variations. Traditional kinship verification models struggle to account for these changes, resulting in decreased accuracy. Accurate kinship verification is crucial for various applications, including forensic investigations, family reunion efforts, and social media analysis to mention some implementations of kinship verification. However, traditional kinship verification models struggle to account for these changes, resulting in decreased accuracy. To tackle this, the objective was to enhance kinship verification by integrating age transformation techniques into a deep learning model. The proposed solution involved applying the LATS (Learnable Age Transformation Synthesis) algorithm to transform images into different age ranges, making familial traits more recognizable. A deep learning model using a Siamese network architecture was trained on the Families in the Wild (FIW) dataset, with age transformations applied at 5, 15, and 30 years to address the model's ability to identify kinship relationships of mother, father and children. The model was evaluated using accuracy, F1-score, and Mean Squared Error (MSE) across different transformation scenarios. The results showed an overall accuracy of 87 %, with the best performance in father-children relationships at a 5-year transformation and in mother-children relationships at a 15-year transformation, demonstrating the model's effectiveness in capturing age-specific familial traits.

Contents

1. Introduction	1
1.1. Problem	5
1.2. Document Structure	7
2. Theoretical Framework	9
2.1. Kinship Verification	10
2.2. Kinship Identification	10
2.3. Kinship Classification	11
2.4. Siamese Network	12
2.5. ResNet (Residual Network)	13
2.6. Triplet Loss Function	14
2.7. Hyperparameter Tuning	15

2.8.	Keras Tuner	15
2.9.	Available datasets	16
2.9.1.	Cornell Kin Face	17
2.9.2.	UB KinFace	17
2.9.3.	KinfaceI and KinfaceII	17
2.9.4.	Family 101	18
2.9.5.	Families in the Wild (FIW)	18
2.9.6.	WVU Kinship Video Database	19
2.10.	Related Work	19
2.10.1.	Methodologies	20
2.10.2.	Comparative Analysis of Current Kinship Verification Methods	23
2.10.3.	Gap Analysis of Current Kinship Verification Methods	29
2.10.4.	Areas for Further Research and Innovation	31
3.	Hypothesis and Objectives	34
3.1.	Hypothesis	35
3.2.	Objectives	36
3.2.1.	Main Objective	36
3.2.2.	Specific Objectives	36
3.2.3.	Deliverables	37
3.2.4.	Scope	39
4.	Methodology	41
4.1.	Experiment Design	44

4.1.1.	Background	44
4.1.2.	Design	45
4.1.3.	Objective 1: Determine the Limitations within the Research Area	45
4.1.3.1.	Literature Review	45
4.1.3.2.	Comparative analysis	46
4.1.3.3.	Identify main gaps	46
4.1.4.	Objective 2: Design a Algorithm Integrating Age Transformation Techniques and Model Tuning	47
4.1.4.1.	Data Processing	47
4.1.4.2.	Siamese Model Input	48
4.1.4.3.	Feature Extraction	48
4.1.4.4.	Training and Loss Function	49
4.1.5.	Objective 3: Evaluate the Metrics of the Proposed Algorithm by Evaluating Scenarios Focusing on Age Transformation and Model Tuning Utilizing the Families in the Wild (FIW) Dataset	50
4.1.5.1.	Evaluation	50
4.1.5.2.	Results	51
4.1.6.	Case Selection	51
4.1.6.1.	Age Transformation	51
4.1.6.2.	Hyperparameter Tuning	52
4.1.6.3.	Convolutional Filter Tuning	52
4.1.6.4.	Dense Layer Units Tuning	52

4.1.7.	Data Collection	53
4.1.8.	Analysis	54
4.1.8.1.	Accuracy	54
4.1.8.2.	F1-score	54
4.1.8.3.	Mean Squared Error (MSE)	55
4.1.9.	Validity	55
4.1.9.1.	Internal Validity	55
4.1.9.2.	External Validity	56
4.1.9.3.	Construct Validity	58
4.1.9.4.	Relation between facial features and kin verification .	58
4.1.9.5.	Age Transformation	58
4.1.9.6.	Model Tuning	58
4.1.9.7.	Use of established techniques	59
4.1.9.8.	Evaluation of metrics	59
4.1.9.9.	Reliability	59
4.1.10.	Limitations	60
5.	Design	62
5.1.	Proposed algorithm	63
5.1.1.	Data preparation	64
5.1.1.1.	Image Loading and Organization into Triplets	67
5.1.1.2.	Dataset Splitting and Batching	68
5.1.2.	ResNet50 embedded model	68
5.1.2.1.	Embedding model	69

5.1.2.2.	Pre-trained Base Model (ResNet50)	69
5.1.2.3.	Additional Convolutional Layers	69
5.1.2.4.	Dense Layers	71
5.1.2.5.	Output Layer and Trainable Layers	71
5.1.3.	Siamese Network Model	71
5.1.4.	Siamese Model Class	72
5.1.4.1.	Distance Layer	75
5.1.4.2.	Loss Function	75
5.2.	Model Compilation	76
5.2.1.	Hyperparameter Tuning	76
5.2.2.	F1-Score method	78
5.3.	Limitations and Requirements	81
5.4.	Implementation	83
6.	Results	84
6.1.	Kinship Verification Model Evaluation	85
6.1.1.	Kinship relation father-children	88
6.1.1.1.	5 years Image Transformation Scenario	88
6.1.1.2.	15 years Image Transformation Scenario	91
6.1.1.3.	30 years Image Transformation Scenario	93
6.1.2.	Kinship relation mother-children	95
6.1.2.1.	Mother-Children 5 Years Image Transformation Scenario	95

6.1.2.2. Mother-Children 15 Years Image Transformation	
Scenario	98
6.1.2.3. Mother-Children 30 Years Image Transformation	
Scenario	100
7. Discussion	103
7.0.1. Father-Children kinship verification analysis	104
7.0.2. Mother-Children kinship verification analysis	107
7.0.3. General results analysis of the proposed model	112
7.0.4. Contributions	117
7.0.5. Limitations	120
8. Conclusions and Future Work	122
8.1. Conclusions	123
8.2. Future Work	126
9. Appendixes	130
9.1. Source Code	131
References	133

List of Figures

5.1. Representation of the proposed model.	64
6.1. Evaluation process of proposed model.	86
6.2. Hyperparameter values for each fold in the father-children 5-year transformation scenario.	90
6.3. Hyperparameter values for each fold in the father-children 15-year transformation scenario.	92
6.4. Hyperparameter values for each fold in the father-children 30-year transformation scenario.	94
6.5. Accuracy across folds for 5-year age transformation in the mother- children relationship.	97
6.6. Accuracy across folds for 15-year age transformation in the mother- children relationship.	99

6.7. Accuracy across folds for the 30-year age transformation in the mother-children relationship.	102
7.1. Best accuracy got on 5-years old transformation for relation Father-Children	113
7.2. Best accuracy got on 15-years old transformation for relation Mother-Children.	114
7.3. Comparison of Father-Mother accuracy through age transformations.	115
7.4. Transformation mother-kid to 5 years old.	116
9.1. Colab Pro configuration.	131

List of Tables

2.1. Comparative Analysis of Kinship Verification Methods	24
3.1. Deliverables in Kinship Verification Research	37
6.1. Hyperparameter values for 5 years old transformation	89
6.2. Average validation metrics across all folds	90
6.3. Hyperparameter values for each fold	92
6.4. Average validation metrics across all folds	93
6.5. Hyperparameter values for each fold	94
6.6. Average validation metrics across all folds	95
6.7. Results of hyperparameter values for each fold 5 years old mother- children kinship verification	96

6.8.	Average validation metrics across all folds for 5-year mother-children age transformation scenario.	98
6.9.	Results of hyperparameter values for each fold in the mother-children 15-year transformation scenario.	99
6.10.	Average validation metrics across all folds for 15-year mother-children age transformation scenario.	100
6.11.	Hyperparameter values for each fold in the mother-children 30-year transformation scenario.	101
6.12.	Average validation metrics across all folds for the 30-year mother-children age transformation scenario.	101
7.1.	Comparison of Father-Children Relationship Accuracy Across Different Kinship Verification Models	106
7.2.	Comparison of Father-Children Relationship Accuracy for Models Using Age Transformation	107
7.3.	Comparison of Mother-Children Relationship Accuracy Across Different Kinship Verification Models	109
7.4.	Comparison of Mother-Children Relationship Accuracy for Models Using Age Transformation	111
7.5.	Comparison of Global Accuracy Across Different Kinship Verification Models	119

Chapter 1

Introduction

The ability to verify kinship relationships through facial features has significant implications in various fields, such as forensics, social media analysis, and family reunification efforts. Despite the advancements in facial recognition technology, it remains a challenging problem, primarily due to the variations in facial features caused by aging. When families are separated due to wars, natural disasters, or adoption, accurately identifying familial relationships can play a crucial role in reuniting individuals with their loved ones. This research addresses the critical challenge of how age-related transformations affect the reliability of kinship verification systems, especially when comparing parents with their younger children. The complexity of facial changes over time, combined with factors such as genetics and environmental influences, makes this task even more complicated.

Different studies have highlighted that traditional kinship verification models often fail to consider the dynamic nature of facial aging, leading to inaccuracies in identifying kin relationships. According to [1], facial features can change significantly over time, resulting in decreased performance when algorithms attempt to match family members across different age ranges. This limitation emphasizes the necessity for developing advanced models that can adapt to age variations. Hence, there's a clear need to address how these age-related changes influence kinship verification to enhance the reliability of these systems.

The main objective of this thesis is to enhance kinship verification by integrating age transformation techniques with deep learning models, aiming to capture familial characteristics across different age ranges more effectively. This integration aims to capture familial face features across different age ranges, making kinship verification

more robust and reliable. The proposed approach seeks to transform facial images into three specific age ranges—5, 15, and 30 years—based on medical evidence that suggests similarities in facial features between parents and children become more apparent when both are within the same age range. By addressing this age gap, the model will be able to identify familial relationships more accurately.

The proposed solution involves using the LATS (Learnable Age Transformation Synthesis) algorithm to transform facial images, thereby capturing more recognizable familial facial features across different age ranges. This transformation, combined with a deep learning architecture based on a Siamese network, enables the model to handle variations in facial features due to aging. To optimize the model’s performance, the `optuna` library is employed to fine-tune hyperparameters, ensuring the most effective configuration for each scenario.

The validation process involves testing on the Families in the Wild (FIW) dataset, one of the largest publicly available datasets for kinship verification. The model’s effectiveness is evaluated using metrics such as accuracy, F1-score, and Mean Squared Error (MSE), providing a comprehensive analysis of its performance.

The contributions of this thesis are as follows:

- **Integration of Age Transformation Techniques:** This work incorporates age transformation techniques into kinship verification, providing a new approach to addressing age-related variations in facial features.
- **Optimization through Hyperparameter Tuning:** The use of `optuna` ensures that the model is optimized for performance.

-
- **Validation through Case Studies:** By applying the model to age transformation scenarios, the research demonstrates the practical utility of age transformation techniques in kinship verification, offering valuable insights for future applications.
 - **Improved Accuracy in Kinship Verification:** The proposed model demonstrates its effectiveness in handling age-related variations in familial facial features, achieving high accuracy in kinship verification across different age ranges.

In summary, this thesis aims to contribute to the field of kinship verification by offering an approach that integrates age transformation techniques with deep learning and also offers more insights for future research in the kinship verification area.

1.1. Problem

is the process of determining familial relationships based on facial features. It plays a crucial role in various applications such as family reunification, forensic investigations, and understanding social connections. However, recognizing kinship through facial features presents unique challenges that are not typically encountered in general face recognition tasks. One of the main challenges is the age gap between family members, as facial features naturally change over time due to aging. This means that the facial resemblance between parents and their children may become less obvious as the years pass, making it harder to identify the kinship relation. Additionally, variations in facial expressions, lighting conditions, and even lifestyle factors can further complicate the verification process, requiring more sophisticated techniques to accurately capture and analyze these familial traits. Therefore, addressing these challenges is essential to improving the accuracy and reliability of kinship verification systems, particularly in situations where accurate identification can have significant personal or societal impacts.

According to Wang, Ding, and Fu, Yun (2019) the generational is a big challenge due to differences in changes corresponding to facial features affected by time. Traditional algorithms failed to take into consideration these age-related transformations on face features, as a result the accuracy decreased when comparing older parents with young children[2]. Research has shown that changes in age impacts directly the reliability of the , one example of age gap is that siblings tend to share more similar facial features compared to their parents where face features are different due to age differences and accuracy is higher for sibling pairs

1.1. Problem

than for parent-child pairs due to the features in faces that changes with time[3].

Cross-generation feature learning is a process in the area that leverages interaction between features from different generations to better understand kinship. The main challenge with this technique is that it requires large and diverse datasets, which in the kin image processing area is a top issue due to the lack of big datasets made of real families[4].

Integrate age transformation into the verification process is another technique to address the issue. By transforming images into ages between 15 and 19 where all family member will have same features gives the model the ability to recognize kinship cross different ages but, it represents a computational challenge due to the time in processing the images[1].

The problem relies on build a robust and accurate model that accounts for age gap between family members and also that address the computational challenge. This work concentrates in use of image processing techniques to transform families photos to a certain age (it is proposed to convert them to 5, 15 and 30) and also combine model tuning methodologies such as to improve the metrics of the algorithm. The hypothesis aims to improve the metrics of the algorithm compared with the conventional image analysis model by combining the mentioned techniques.// In summary, involves determining familial relationships based on facial features, but it faces challenges such as age gaps between family members. Research indicates that traditional algorithms often fail to account for age-related changes in facial features, leading to reduced accuracy, particularly when comparing older parents with young children. Sibling pairs typically have higher

verification accuracy due to their more similar facial features. Cross-generation feature learning and age transformation techniques have been proposed to address these issues. However, these methods require large datasets and present computational challenges. This work focuses on using image processing to transform family photos to certain ages and applying model tuning methodologies to enhance the accuracy of kinship verification algorithms. The goal is to improve the algorithm's performance compared to conventional models by integrating these advanced techniques.

1.2. Document Structure

This document aims to present the research and findings on kinship verification using age transformation techniques and deep learning models. Each section is structured to provide a clear understanding of the study, from the foundational concepts to the final conclusions.

The Theoretical Framework chapter introduces the key theoretical concepts that are necessary to understand this research. It also provides an overview of the related work, showcasing the existing approaches and methods that have been developed in this field.

In the Hypothesis and Objectives chapter shows the main hypothesis of this research, along with a clear description of the objectives and what this work aims to deliver. This section also outlines the scope of the study.

The Methodology chapter highlights the experimental setup, explaining the

1.2. Document Structure

design of the experiment using a test case design methodology. This section gives a detailed look into the process behind the research.

Next, the Design chapter provides insight into how the implementation was carried out. It discusses the technical aspects of the work, including the design choices and how everything was put together.

The Results chapter presents the outcomes of the experiments, showing the data collected during the study. The Discussion chapter then dives into analyzing these results, helping to make sense of what they mean and how they contribute to the overall research.

Finally, Conclusions and Future Work wraps up the document with the conclusions from the research, highlighting the main findings. It also includes suggestions for future work.

Theoretical Framework

This section explores key concepts such as , identification, and classification, which are essential for analyzing relationships between individuals. It also dives into the concepts of facial vectors, architectures, ReLU activation functions, and Multi-Layer Perceptrons (MLPs). The aim of this section is to provide a comprehensive understanding of how these components interact and function within a model.

2.1. Kinship Verification

The objective behind is to identify whether two persons have a kin relationship[5] where a DNA analysis is the most effective way to accomplish this process but, sometimes is not applicable (like looking for missing people) and also it comes with a cost.

Human face is important and part of [6], that is one of the reasons why this area has become popular in the image analysis using computational models, making facial analysis a valuable alternative for [6]. The human face carries a wealth of information that reflects genetic traits, making it possible to identify familial connections based on facial similarities. This has led to increased interest in using computational models and image analysis techniques to study kinship relationships.

2.2. Kinship Identification

Kinship identification is the process of determining the specific type of kinship relationship between two individuals[7]. Unlike general kinship verification, which only establishes whether a familial connection exists, kinship identification goes a

2.3. Kinship Classification

step further by specifying the exact relationship, such as mother-son, mother-daughter, or sibling connections. This task is more complex because it requires the model to distinguish between various familial ties, which may have subtle differences in facial features. The ability to accurately identify the type of kinship is crucial for applications that require detailed family mapping, such as genealogical studies, family reunification efforts, and personalized social media experiences. For instance, recognizing whether an image shows a mother-son or a mother-daughter relationship involves detecting nuanced differences in facial structures, expressions, and other inherited traits that are characteristic of these specific relationships. By focusing on identifying these particular connections, kinship identification helps provide a more comprehensive understanding of how family traits are passed down through generations.

2.3. Kinship Classification

Kinship classification aims to identify the specific family to which an individual belongs. Unlike kinship verification and identification, which focus on determining familial relationships between pairs of individuals, kinship classification deals with assigning a person to a particular family group[7]. This process involves analyzing facial features and patterns that are shared among family members, allowing the model to classify individuals into their corresponding family. This area of research is particularly useful in cases such as finding the families of missing children or reuniting individuals with their biological families after being separated due to adoption. For

instance, in situations where children have been displaced or separated from their families, kinship classification can help match them to their relatives by comparing their facial features with those of potential family members.

2.4. Siamese Network

To compare two inputs and determine their similarity, Siamese networks are the right type of neural network architecture where a Siamese Network has two identical subnetworks that share the same weights and parameters. The idea behind this design is that each subnetwork processes one of the input images to produce an output vector; then, the two vectors are compared using a distance metric to determine the similarity between the two inputs.

The Siamese networks learn discriminative feature space by implementing a contrastive loss function during training. The Contrastive loss function will minimize the distance between embeddings of similar pairs and maximize the distance between embeddings of dissimilar pairs. Due to this feature, Siamese networks are ideal for managing a verification task where the main objective is to determine if the two inputs are from the same class. The idea of Siamese networks was introduced in 1994 by Bromley et al. in their seminal paper[8].

The initial use of Siamese networks was applied to signature verification, but later it was used in numerous applications in diverse fields of pattern recognition like where it proves to be a powerful tool due to its capacity to capture features that indicate kin relationships[7].

The versatility of the Siamese networks allows them to meet specific needs of different applications, which makes them a valuable tool in the machine learning area. For example, the study by Lu et al. (2017) is a good example of deep learning metric for due to the capacity of Siamese networks to learn meaningful and discriminative features. This is achieved by mapping the facial images of related individuals closer together in the feature space, improving the accuracy while verifying kinship[9].

2.5. ResNet (Residual Network)

Residual Networks is a deep neural network architecture that manages the training process of deep networks, it works with residual connections that allow the network to learn residual functions instead of approximating the underlying mapping.

That means that, instead of trying directly to learn from the function $H(x)$ where it is the desired output of a layer, residual connections learn from the difference between $H(x)$ and x , expressed as $F(x)=H(x)-x$. Then the output of a layer becomes the sum of the original input x and the residual function $F(x)$, returning $H(x)=F(x)+x$.

ResNet50 is a variant of ResNet architecture consisting of 50 layers. Two of its main features are the balance between model complexity and computational cost. Also, the ResNet50 is a popular choice for different computer tasks like image classification, object detection, and feature extraction.

The ResNet concept is introduced by the authors in their seminal paper, where

they demonstrate the effectiveness in image recognition task[10].

2.6. Triplet Loss Function

Triplet loss is a loss function commonly used in Siamese neural networks for learning embeddings where it learns a feature space where the distance between embeddings of similar instances is minimized, while the distance between embeddings of dissimilar instances is maximized[1].

Function is $\text{triplet_loss} = \max(0, d(a, p) - d(a, n) + \alpha)$ where:

$d(a, p)$ is the distance between the anchor a and the positive example p in the feature space,

$d(a, n)$ is the distance between the anchor a and the negative example n in the feature space, and

α is a positive margin to ensure effective learning without trivial solutions.

The function $\max(0, x)$ is known as the hinge loss function.

It ensures penalization only occurs when $d(a, n)$ is less than $d(a, p) + \alpha$.

For facial , triplet loss helps in effectively discriminating between kin and non-kin faces. Abbas and Shoaib[1] utilized age transformation and along with triplet loss for .

2.7. Hyperparameter Tuning

Hyperparameters are parameters applied to the learning process of the machine learning models, being external to the model where these cannot change during the learning process but are used during learning. These parameters are set before the training, playing a crucial role in controlling the behavior of the algorithm and the structure of the model[10].

They can be categorized into two main types: model and algorithm. Model hyperparameters set the architecture of the model, such as the number of layers in a neural network, the number of units per layer, and the activation function that will be used. Algorithm hyperparameters work directly with the learning process. Some examples are the learning rate, batch size, the number of epochs, and optimization techniques. The use of these parameters will help to improve the performance of the model since they can affect the ability of learning[11].

The main objective of hyperparameters is to find the best combination that maximizes the model's performance on a validation set. Some techniques for hyperparameter tuning include grid search, random search, and evolutionary algorithms. Effective tuning can lead to improvements in the model accuracy, robustness, and efficiency[7].

2.8. Keras Tuner

Keras Tuner is a library designed to implement easily the process of used in machine learning models where the hyperparameters play a crucial role in controlling

the behavior of the algorithm and the structure of the model[10]. They aim to find the best combination of hyperparameters to optimize the performance of the machine learning process. One of the main features of Keras Tuner is that it can be easily integrated with the Keras models. The idea is that users can define their models in Keras and implement Keras Tuner to explore different hyperparameter combinations.

The integration between Keras models and the Keras Hyperparameters allows a smooth workflow where users can focus on the model while Keras Tuner takes ownership of the hyperparameter management.

Multiple search algorithms are supported, like Random Search, Bayesian optimization, and Hyperband. This work focuses on Random Search, which is a simple and effective method that explores the hyperparameter space by selecting different combinations[12].

2.9. Available datasets

When studying , different datasets are available for research proposes. Each dataset offers unique features and challenges. This section provides a brief overview of the modes commonly used datasets in literature where it is highlighted their characteristics. It is important to understand the datasets for better understanding of available data and shape the development of future models.

2.9.1. Cornell Kin Face

It is the first database created for , it was created by Fang, Tang, Snavely, and Chen [13] from the University of Cornell. The database has 150 pairs of a total of 300 images with variations in demographic attributes. The database contains pairs of relations like father and son in 40%, father and daughter 13%, mother and son and mother and daughter in a 26%. One big issue this database contains is the poor amount of images to train the models. The size of the images is 100px x 100px, and all images are in a .jpeg format.

2.9.2. UB KinFace

This database was proposed by Shao, Xia, and Fu[14] and contains a total of 600 images. The images belong to 200 families, which have images of a child, a young parent, and an old parent. It was the first dataset to include variations in age between children and their parents. It contains 91 images for father and son relation, 79 images for father and daughter, 15 images for mother and son and 21 images for mother and daughter. Images are in 127px x 100px and are in the .jpeg format.

2.9.3. KinfaceI and KinfaceII

This database is proposed by Lu, Wang, and Shan[15]. Both datasets include images from the web captured under uncontrolled environments meaning gestures, demographic attributes, light, background, expressions and partial occlusions are natural from the original picture. This database contains 4 kin relations, 156 relations

for father and son, 134 for father and daughter, 116 for mother and son, and 127 mother and daughters of kinship pairs. The KinfaceII was enlarged with celebrities images as well their children or parents making this dataset grown in 250 images for each kin relation (F-D, F-S, M-D, M-S). The format of the images are .jpeg and the size is 64px x 64px.

2.9.4. Family 101

This dataset was proposed by Fang [16] after proposing Cornell’s dataset. The dataset contains 14,816 images with 101 different family trees where 206 are nuclear families, 607 are individuals images. It was created from public families, and it has four kinship relations where 213 are father and son, 147 are father and daughter, 184 are mother and son and 148 belongs to the relation between mother and daughter. It is the first large-scale dataset of families across several generations, the image size is 150px x 150px and are in .jpeg format.

2.9.5. Families in the Wild (FIW)

This is the biggest dataset for kinship image analysis. This database contains 300 families where 1828 are family members and, 10255 are the images in total. The number of images per member differs a lot, an example is not the same amount of images that exist for a father than the amount of images his son has. Majority of the members has less than 10 image examples more than 250 people has only 1 or 2 image example, and a few ones have more than 20 examples[17].

2.9.6. WVU Kinship Video Database

The dataset contains video clips of individuals and kin relations, where data is divided between testing and training. The dataset for each individual contains frames extracted from the videos, where it has 141 videos for training and 214 videos for testing. It has seven types of kin-pair relations which includes mother and daughter in 21.28%, mother and son 7.80%, father and daughter 16.31%, father and son 21.28%, brother and brother 11.34%, sister and sister 13.47% and finally brother and sister 8.52%. The dataset has a lack of cross gender relations such as mother and son and brother and sister. It also contains extreme pose variations, making the underlying data distribution highly complex to learn.[18].

2.10. Related Work

This section presents a review of current state-of-the-art work related with . It explores various approaches and methodologies that have been developed to address the complexities of identifying familial relationships through facial recognition. Existing and recent literature in the field of are briefly explained to understand key areas for the foundation of this research such as integration of deep learning techniques, the use of age transformation methods, and the application of advanced neural network architectures such as and . It also examines studies that have utilized the dataset, providing insights into the benchmarks and evaluation metrics used.

2.10.1. Methodologies

Research in the area has gained popularity with the increasing use of machine learning techniques for face recognition. In recent years, more researchers have shown interest in the unique challenges that presents. Various studies have explored different methods and approaches to tackle this problem.

Liang et al. [19] addressed the issue by introducing a weighted graph embedding-based metric learning approach that captures the relational structure of facial features within a family. Their method effectively analyzed the complex relationships between facial characteristics in family members.

Dong, Pun, and Zhang [4], in their work on cross-generation feature interaction learning for , aimed to bridge the generational gap by enhancing the model’s ability to recognize kinship across different ages. Although their approach showed promising results, it required extensive parameter tuning and was prone to overfitting, especially when applied to smaller or less diverse datasets.

Zheng and Hu [6] proposed a novel technique that compared input images against a reference list of known kin relationships, offering an alternative way to enhance the accuracy of kinship verification. In another study, Zheng et al. [20] introduced age-uniform feature learning for , which aimed to standardize the ages of family members in images, making the verification process more efficient and accurate.

Dehshibi and Shanbehzadeh [21] explored the use of cubic norm and kernel-based bi-directional PCA to incorporate age-awareness into , addressing the age variation challenge. Meanwhile, Wang et al. [5] worked on cross-generation using a sparse discriminative metric, enhancing the model’s ability to handle age differences but

requiring careful parameter tuning and significant computational resources.

Laiadi et al. [22] approached by utilizing discriminative subspaces of color components, leveraging color information to improve the verification process. Similarly, Sellam and Azzoune [23] introduced the neighborhood min distance descriptor for , focusing on minimizing the distance between facial features of related individuals.

Liang et al. [24] presented a method using deep relational features to verify kinship, emphasizing the role of deep learning techniques in this field. Their approach effectively extracted and compared relational features to improve accuracy.

El Houda Bouakal et al. [2] proposed a system based on color space analysis, aiming to enhance accuracy by examining different color spaces in facial images. Zhu et al. [3] introduced distance and direction-based deep discriminant metric learning, which focused on improving the model’s discriminative power by considering the distance and direction between facial features.

Lu, Hu, and Tan [9] applied discriminative deep metric learning for face and , leveraging deep learning techniques to boost verification accuracy. Sinha, Vatsa, and Singh [25] developed FamilyGAN, a generative adversarial network designed to generate kin face images, thereby improving through synthetic data augmentation.

Dornaika, Arganda-Carreras, and Serradilla [26] explored transfer learning and feature fusion, combining pre-trained models with features from multiple sources for . Oruganti et al. [27] introduced a deep learning model using childhood images, focusing on improving accuracy by leveraging images at different ages.

Abbas and Shoaib [1] proposed using age transformation combined with a Siamese

network, aiming to improve verification by matching images of individuals at similar ages. Xu and Shang [28] introduced a method that used structured similarity fusion to enhance kinship measurement on facial images.

Zhang et al. [29] explored supervised contrastive learning for facial kinship recognition, while Qin et al. [30] presented tri-subject kinship verification to understand family dynamics by considering relationships between three individuals.

Dahan, Keller, and Mahpod [31] introduced a kin-verification model using multi-set learning and local features, and Goyal and Meenpal [32] focused on multi-level dictionary pair learning for images of varying resolutions in . Zhao et al. [33] proposed learning a multiple kernel similarity metric, using multiple kernels to enhance verification.

Yan, Lu, and Zhou [34] presented a prototype-based discriminative feature learning technique, while Li et al. [35] developed Kinnet, a fine-to-coarse deep metric learning approach. Wang et al. [7] introduced kinship identification through joint learning using ensembles.

He et al. [10] contributed by introducing deep residual learning for image recognition, showing potential in kinship verification. Taigman et al. [36] proposed DeepFace to bridge the gap to human-level face verification, while Schroff et al. [11] introduced FaceNet, a unified embedding for face recognition and clustering.

Significant progress has been made in the field of , there are still challenges, such as the need for larger datasets, computational demands, and sensitivity to image quality. This research contributes to the field by integrating age transformation techniques with model tuning methodologies, aiming to improve kinship verification

across different age groups. The proposed algorithm is validated using the dataset, addressing key gaps and enhancing model performance.

2.10.2. Comparative Analysis of Current Kinship Verification Methods

Kinship verification is a challenging task due to the complexities associated with facial variations across different age groups. Several methods have been proposed to address these challenges, each with unique strengths and weaknesses.

The following table summarizes the key methods, highlighting their approaches, strengths, and weaknesses. This comparative analysis will serve as a foundation for understanding how the proposed model fits within the broader context of kinship verification research.

The objective of this analysis is to understand how each method tackles the inherent complexities of kinship verification, such as age transformations, variations in pose, lighting conditions, and the ability to generalize across different family structures.

Table 2.1: Comparative Analysis of Kinship Verification Methods

Reference	Method	Strengths	Weaknesses
[21]	Cubic Norm and Kernel-based Bi-directional PCA	▪ Incorporates age information ▪ Effective dimensionality reduction	▪ May struggle with real-time applications due to computational complexity ▪ Performance might be dataset-dependent
[5]	Cross-Generation Kinship Verification with Sparse Discriminative Metric	▪ Effective in handling cross-generation variations ▪ Sparse representation reduces overfitting	▪ Sparse models can be sensitive to noise ▪ Requires careful tuning of hyperparameters

Continued on next page

Reference	Method	Strengths	Weaknesses
[6]	Kinship Verification via Reference List Comparison	▪ Simple and interpretable ▪ Effective for small datasets	▪ Limited scalability to large datasets ▪ May not capture complex kinship relationships
[19]	Weighted Graph Embedding-Based Metric Learning	▪ Captures complex relationships through graph embeddings ▪ Robust to variations in pose and expression	▪ Computationally intensive ▪ Requires high-quality graph construction

Continued on next page

Reference	Method	Strengths	Weaknesses
[4]	Cross- Generation Feature Interaction Learning	▪ Handles cross-generation variations effectively ▪ Learns robust feature interactions	▪ Potentially high model complexity ▪ Requires substantial training data
[20]	Age-uniform Feature Learning	▪ Uniform features reduce age-related discrepancies ▪ Improved generalization across age groups	▪ May not perform well with non-uniform age distributions ▪ Requires careful feature extraction

Continued on next page

Reference	Method	Strengths	Weaknesses
[22]	Discriminative Subspaces of Color Components	▪ Leverages color information effectively ▪ Simple implementation	▪ Limited to color images ▪ May struggle with grayscale or low-quality images
[23]	Neighborhood Min Distance Descriptor	▪ Effective for fine-grained kinship relationships ▪ Robust to small variations	▪ Sensitive to outliers ▪ Limited scalability

Continued on next page

Reference	Method	Strengths	Weaknesses
[24]	Deep Relational Features	▪ Powerful feature extraction with deep learning ▪ High accuracy on large datasets	▪ Requires extensive computational resources ▪ Potential overfitting on small datasets

2.10.3. Gap Analysis of Current Kinship Verification Methods

The models present on the state-of-the-art in the kinship verification area presents some gaps that authors requires to address in further investigations. Here some of the gaps found after an analysis of them:

- **High Computational Complexity:** Many methods, such as those using deep learning and complex models (e.g., Cubic Norm and Kernel-based Bi-directional PCA, Weighted Graph Embedding-Based Metric Learning), are computationally intensive and not suitable for real-time applications.
- **Sensitivity to Noise and Outliers:** Techniques like sparse discriminative metric learning and neighborhood min distance descriptor are sensitive to noise and outliers, which can affect their robustness and accuracy.
- **Dependency on Large Datasets:** Several methods (e.g., Discriminative Deep Metric Learning, Deep Relational Features) require extensive training datasets to perform well, limiting their applicability when large labeled datasets are unavailable.
- **Complex Model Architectures:** Methods such as Distance and Direction Based Deep Discriminant Metric Learning and Cross-Generation Feature Interaction Learning have complex architectures, making them difficult to implement and maintain.
- **Limited Generalizability:** Many approaches, including age-specific

methods like Age-uniform Feature Learning, may not generalize well across diverse datasets or varying age distributions.

- **Dependence on High-Quality Inputs:** Methods that rely on color components or specific image qualities (e.g., Discriminative Subspaces of Color Components, Color Spaces Analysis) are limited when dealing with grayscale or low-quality images.
- **Stability and Training Issues:** GAN-based methods like FamilyGAN can be unstable during training, requiring careful tuning and substantial computational resources.
- **Complex Feature Fusion and Selection Processes:** Techniques that involve feature fusion (e.g., Transfer Learning and Feature Fusion) or require careful prototype selection (e.g., Prototype-Based Discriminative Feature Learning) can be complex and time-consuming.
- **High Computational Costs:** Many advanced methods incur high computational costs, limiting their practicality for widespread or real-time use.

While significant advancements have been made in the field of kinship verification, the current methods still face a range of challenges that limit their effectiveness and practicality. Issues such as high computational complexity, sensitivity to noise, dependence on large datasets, and difficulties in generalizability hinder the application of these models in real-world scenarios. Moreover, the

reliance on high-quality inputs and the complexity of model architectures further restrict their adaptability across diverse datasets and conditions. Addressing these gaps is essential for developing more efficient, scalable, and accurate kinship verification models that can be applied across various contexts.

2.10.4. Areas for Further Research and Innovation

Some areas for further research and innovation are mentioned on the available research studies. This analysis groups the most mentioned areas for improvement based on the consulted work in which this thesis was based.

- **Improving Computational Efficiency:** Developing more efficient algorithms and models that reduce computational burden without compromising accuracy. This includes simplifying model architectures and optimizing computational processes.
- **Enhancing Noise Robustness:** Creating techniques that are more robust to noise and outliers, improving the reliability and accuracy of kinship verification methods.
- **Scalability with Limited Data:** Innovating methods that perform well even with smaller datasets, through techniques such as data augmentation, transfer learning, and few-shot learning.
- **Generalization Across Diverse Datasets:** Ensuring that models generalize well across diverse datasets and varying conditions, such as different age distributions and image qualities.

- **Versatile Feature Extraction:** Developing automated and accurate feature extraction methods that work well across various image types, including grayscale and low-quality images.
- **Stabilizing GAN Training:** Enhancing the stability of GAN training processes to ensure consistent performance and reducing the computational requirements for training GANs.
- **Simplifying Feature Fusion Processes:** Simplifying the processes involved in feature fusion and selection to make these techniques more accessible and less time-consuming.
- **Reducing Computational Costs:** Innovating ways to reduce the computational costs associated with advanced kinship verification methods, making them more practical for real-time and widespread use.
- **Automating Hyperparameter Tuning:** Developing automated hyperparameter tuning processes to improve model performance without extensive manual intervention.
- **Leveraging Unlabeled Data:** Utilizing semi-supervised or unsupervised learning techniques to make use of unlabeled data, thereby enhancing the training process and model robustness.

Different areas are identified for further research and innovation highlight the need for advancements that address the limitations and challenges currently faced in kinship verification. Improving computational efficiency, enhancing noise

2.10. Related Work

robustness, and ensuring scalability with limited data are critical steps toward making these models more practical for real-world applications. Additionally, focusing on generalization, versatile feature extraction, and stabilizing training processes will contribute to developing more adaptable and reliable models.

Hypothesis and Objectives

This chapter presents the principal components that guide the research conducted in this thesis. It begins by outlining the hypothesis, which addresses the inherent challenges of kinship verification when faced with significant variations in age, pose, and facial features. Then, it presents details the objectives of the research and finally, outlines the deliverables associated with each specific objective and concludes with a summary of the scope and limitations of the research.

3.1. Hypothesis

Variations in age, pose and face features are some challenges when accurate kinship verification from images comes into the picture. Traditional models often struggle with these differences and leaves them out of the scope during developing. The hypothesis proposes that using image processing techniques to age-transform family images, combined with model tuning methodologies like Keras Tuner, will improve kinship verification metrics compared to conventional models. This approach aims to provide a more consistent basis for verification, enhancing accuracy and reliability. The hypothesis for this research effort reads as follows:

The use of image processing techniques to transform images of families to a certain age, combined with model tuning methodologies such as Keras Tuner, will improve the metrics of the kinship verification algorithm compared with the conventional image analysis model.

3.2. Objectives

The approach present by this work aims to provide a solution for the aging challenge found in facial feature differences when working with . Significant age differences between family member tend to complicate the verification process. A deep learning algorithm for that mixes age transformation method with hyperparameter tuning techniques is presented to face the aging challenge when face features changes. The next objectives are guide to fulfill all requirements in the kinship verification research area.

3.2.1. Main Objective

To propose a deep learning algorithm that addresses the analysis and recognition that improves metrics through the integration of age transformation techniques applied to the family images and model tuning methodologies when significant age differences between family member complicates the verification process.

3.2.2. Specific Objectives

1. To determine the limitations within the research area with the aim of finding areas for improvement.
2. To design a algorithm integrating age transformation techniques and model tuning that enhances the algorithm capability to identify familiar relationships.

3.2. Objectives

3. To evaluate the results of the proposed algorithm by implementing metrics applied to scenarios that focus on age transformation and model tuning using the Families in the Wild (FIW) dataset.

3.2.3. Deliverables

This section groups the key deliverables associated with each specific objective defined in this work. Each deliverable is aligned with the objectives to ensure that the research objectives are comprehensively addressed and that the outcomes are clearly documented.

Table 3.1: Deliverables in Kinship Verification Research

Objective	Deliverable	Description
Identify limitations within the kinship verification field for improvement.	Comparative Analysis Report	Detailed comparison of current kinship verification methods, including strengths and weaknesses.
	Gap Analysis Report	Documentation of main gaps in existing kinship verification methods, identifying areas requiring further research and innovation.
<i>Continued on next page</i>		

3.2. Objectives

Objective	Deliverable	Description
Design a kinship verification algorithm integrating age transformation techniques and model tuning to enhance the ability to identify familial relationships.	Preprocessed Dataset	Cleaned and preprocessed subset of the Families in the Wild (FIW) dataset. Includes quality selection, resizing, normalization, and age transformation using the Life Span Age Transformation Synthesis (LATS) technique.
	Thesis Document (Model Implementation Report)	Detailed design of the kinship verification algorithm, including the Siamese Network Architecture, feature extraction methods, and triplet loss function implementation. Also includes the hyperparameter tuning process using Keras Tuner.
	Source Code of the Proposed Model Implementation	Complete source code of the kinship verification algorithm.
Evaluate the proposed algorithm using evaluation metrics applied to scenarios with age transformation and model tuning using the Families in the Wild (FIW) dataset.	Thesis Document (Training and Evaluation Report)	Documentation of the training process of the Siamese model, including evaluation results using accuracy, F1-score, and Mean Squared Error (MSE) metrics, with performance analysis presented through charts, graphs, and tables.
<i>Continued on next page</i>		

3.2. Objectives

Objective	Deliverable	Description
	Thesis Document (Analysis, Results, and Conclusions)	Comprehensive analysis of the experiment results, discussion of the findings and implications, conclusions, and recommendations for future work.
	Paper Draft	Includes the abstract, introduction, methodology, results, and conclusions for a potential research paper submission.

3.2.4. Scope

The scope of this thesis is defined by the objectives set to address the challenges in the kinship verification area. This work aims to implement a deep learning algorithms that incorporate age transformation techniques to improve the accuracy and efficiency of kinship verification models. The scope of this work covers the following key areas, each aligned with the objectives outlined in this thesis:

1. Development of a deep learning algorithm for kinship verification that integrates age transformation techniques and model tuning methodologies.
2. Application of the proposed algorithm to a subset of the Families in the Wild (FIW) dataset, focusing on scenarios with significant age differences between family members.
3. Evaluation of the algorithm's performance using established metrics such as

3.2. Objectives

accuracy, F1-score, and Mean Squared Error (MSE).

4. Comparative analysis of the proposed algorithm against conventional kinship verification models, identifying areas of improvement and highlighting the advantages of the new approach.
5. Documentation of the research process, including the design, implementation, and evaluation of the algorithm, to provide a comprehensive guide for future studies in the kinship verification area.

By working on these areas, the research aims to make a contribution to the field of kinship verification, offering insights into the application of age transformation techniques in deep learning.

Methodology

To investigate the hypothesis, a case study was conducted to assess the metrics of the proposed deep learning algorithm. A case study is a research method used to investigate phenomena within their real-life context, particularly when the boundaries between phenomenon and context are not clearly evident. This method is particularly useful for exploring, describing, or explaining complex software engineering issues through detailed contextual analysis of a limited number of events or conditions and their relationships[37].

A case study contains different areas that need to be followed: Background, Design, Case Selection, Data Collection, Analysis, Validity and Limitations. Here is a summary of each area:

■ **Background**

- The research is structured around key questions addressing challenges in kinship verification, particularly when there's an age gap between family members. It investigates how current models manage aging-related changes in facial features and explores the integration of age transformation techniques with model tuning in a deep learning framework. The study also considers how to validate the proposed algorithm using the Families in the Wild (FIW) dataset.

■ **Design**

- The experimental design is structured to meet the research objectives. It involves selecting appropriate methods for data extraction, processing, and pre-processing, as well as implementing techniques to achieve final

results and evaluating the outcomes.

■ **Case Selection**

- To enhance the kinship verification algorithm, the model is trained under different scenarios, incorporating age transformation and hyperparameter tuning. The selection of cases involves transforming images to different age ranges and adjusting model parameters to optimize performance.

■ **Data Collection**

- The research utilizes the FIW dataset, which contains a vast collection of family images across generations. A subset of families is selected based on specific criteria to ensure data quality. The data is processed to remove low-quality images and organized into a simplified structure to facilitate model training.

■ **Analysis**

- The model is evaluated under different scenarios that include age transformation and model tuning. Metrics such as accuracy, F1-score, and MSE are used to measure performance, with results presented for easy comparison.

■ **Validity and Limitations**

- The validity of the research is assessed in terms of internal, external, construct, and reliability. Internal validity is ensured through

cross-validation, while external validity acknowledges dataset limitations. Construct validity is established through the use of established techniques and metrics. Reliability is achieved through multiple validation techniques and recognized metrics. The limitations of the FIW dataset, such as unbalanced and low-quality images, are also discussed, highlighting the need for manual data cleaning and the challenges of image transformation.

4.1. Experiment Design

The Experiment Design section outlines the methodological approach used to validate the proposed kinship verification algorithm. This section details the dataset preparation, age transformation techniques, and evaluation metrics applied, ensuring a robust and replicable framework.

4.1.1. Background

To structure the experiments, the following questions were formulated to work as the backbone structure:

- What are the common challenges faced in when there is a significant age gap between family members?
- How do current models handle changes in facial feature due to aging?
- How can the integration of age transformation techniques with model tuning

4.1. Experiment Design

methodologies be effectively implemented in a deep learning framework?

- How can the proposed algorithm be validated and tested using the Families in the Wild (FIW) dataset?

The guiding questions will help on the design, conducting and analyze process of the experiments ensuring the right path to approach this work objectives.

4.1.2. Design

The experimental design addresses each objective. It details the process and techniques to be used during the experimental phase. It involves selecting the right process to extract information from the current state of the art, process the collected data and pre-process it. Also, it covers the techniques to implement in order to obtain final results and the evaluation methodology.

4.1.3. Objective 1: Determine the Limitations within the Research Area

4.1.3.1. Literature Review

A literature review is a method to group information and highlight insights from existing research that provides a solid foundation to understand the current state of the art. It is a good technique to guide new investigation, identify gaps and areas that require further investigation. By reviewing different studies, it can be identified the latest research and methodologies used in the desired field to investigate[10].

A comprehensive review of existing research in the field of was performed, focusing

4.1. Experiment Design

on the main techniques, datasets and evaluation metrics. These are the request questions in which the literature review extraction was based:

- What are the main facial recognition techniques used in ?
- What is the configuration implemented to perform the accuracy tests?
- Which datasets were used and its main characteristics?

Academic databases such as SCOPUS, IEEE Xplore, Google Scholar, and Springer was used to find relevant papers and articles.

4.1.3.2. Comparative analysis

A comparative analysis of methods was conducted to identify the main challenges and limitations that this area presents. With this analysis:

- Strengths of common techniques would be identified.
- Weaknesses of common techniques would be identified.

This comparison will provide a deep understanding of limitations and challenges in the area and how they can be address.

4.1.3.3. Identify main gaps

The main objective is to reveal areas where current methods fails based on the comparative analysis. By identifying the gaps, areas that require further research and innovation was pointed out. By highlighting these gaps, new research can be proposed to focus on these areas and develop new solution like the one proposed

in this work, where merging age transformation techniques and methods like will provide an innovative solution to .

4.1.4. Objective 2: Design a Algorithm Integrating Age Transformation Techniques and Model Tuning

4.1.4.1. Data Processing

The database that was use is Families in the Wild (FIW) which is a large scale database of kinship images designed to research on the and recognition areas. The FIW database consist of an extensive collection of family images cross multiple generations. It includes family relationships such as parent-child, siblings, which enables the development of kinship algorithms[17]. Since the database has 300 families, was used a subgroup of 144. Selected families are based on amount of pictures on each family member (need to have more than 1 picture in each member).

Preprocessing steps such as manual selection of quality images, resize and normalization was conduced to prepare them for the model. Age Transformation was implemented by using the Life Span Age Transformation Synthesis (LATS) technique to transform images to target ages of 5, 15, and 30 years. This age transformation technique was used due to its ability of addressing variations in age images[1].

4.1.4.2. Siamese Model Input

As input for the Siamese model was used a triplet vector where entries are anchor image, positive image, and negative. The anchor image was varied between mother and father. When the anchor is the mother, the real father was replaced with a father from a different family. Similarly, when the father is the anchor, the mother was replaced with a mother from a different family. Multiple triplet sets was created from the selected families to ensure a comprehensive training dataset. Triples was used due to their capability of training deep networks by helping them to learn embeddings that discriminate between kin and no kin relationships by maximizing the distances between anchor-negative and minimizing distances between anchor-positive [9].

4.1.4.3. Feature Extraction

The feature extraction was in charge of the Architecture, where two identical subnetworks share weights. Each subnetwork was capable of extracting features from the input triplet vectors. Each subnetwork is a ResNet50 which is pretrained on ImageNet dataset. ResNet50 is a variant of the RestNet network. ResNet was used because it offers strong performance cross different computer vision task because it learns hierarchical representation, and it can skip connections through its residual connections between layers, making it effective and facilitate the training process[17].

4.1.4.4. Training and Loss Function

A function was implemented by the Siamese models to leverage feature recognition across images, where the proposed Siamese model process as input a triplet feature vector. This loss function technique is utilized in diverse studies on such as those by Zheng et al. [6], Liang et al. [4], and Lu et al. [12] where the function improves the accuracy and reliability.

The triplet loss function enhances the model's ability to learn discriminative embeddings by minimizing distances between images of the same family members (anchor-positive pairs) and maximizing distances between images of unrelated people (anchor-negative pairs). Here is the triplet loss function that was used:

$$L(A, P, N) = \max(\|f(A) - f(P)\|^2 - \|f(A) - f(N)\|^2 + \text{margin}, 0)$$

- A , P , and N represent the anchor image, positive image, and negative image.
- $f(\cdot)$ is the feature embedding function learned by the Siamese network.
- $\|\cdot\|^2$ is the squared Euclidean distance between the embeddings of the anchor and positive images, and between the anchor and negative images.
- margin is a hyperparameter that specifies the separation threshold or margin between the embeddings of positive and negative pairs.
- $\max(\cdot, 0)$ makes sure that the loss is only computed when the distance between the embeddings of the anchor and positive pairs is greater than the distance between the embeddings of the anchor and negative pairs plus the margin.

In the training phase, the model will computed iteratively the loss using the defined function, and then uses this loss to adjust the model's weights through backpropagation and Adam optimization algorithm. Adam optimization was used due to its effectiveness in also, the robustness and adaptability of this algorithm achieves high accuracy in recognizing familiar relationships due to its ability to learn discriminative features[9, 19].

4.1.5. Objective 3: Evaluate the Metrics of the Proposed Algorithm by Evaluating Scenarios Focusing on Age Transformation and Model Tuning Utilizing the Families in the Wild (FIW) Dataset

4.1.5.1. Evaluation

To evaluate the model's performance, these was the metrics to use:

- Accuracy
- F1-score

These metrics was applied since they are the most commonly used in state-of-the-art research. To ensure that the tuning process is not causing overfitting on the model, the Mean Square Error (MSE) was used for evaluation.

4.1.5.2. Results

Evaluation results was present using different methods like graphs, tables and chats as needed. Additionally, it was summarized the key findings by using the evaluation results. Also, it was discussed the improvements made by the proposed approach, challenges and future work.

4.1.6. Case Selection

To improve the metrics of the algorithm, the model was trained on different scenarios that cover the aging transformation on images and the . In this section, the case selections are explained.

4.1.6.1. Age Transformation

To transform the images, the Life Span Age Transformation Synthesis (LATS) technique was used[38]. Selected families were transformed into this age ranges:

1. Images are transformed into age of 5.
2. Images are transformed into age of 15.
3. Images are transformed into age of 30.

As input for the Siamese model was used a triplet vector where entries are anchor image, positive image and negative. The anchor image was varied between mother and father. When anchor is the mother, the real father was replaced with a father from a different family. Same way, when father is the anchor, mother was replaced

with a mother from a different family. In this way, one portion of reliability was cover in the model. The children images was the positive entry for all cases.

4.1.6.2. Hyperparameter Tuning

During training, the model was tuned using the Keras Tuner. The search strategy was Random Search applied to the . The maximum amount of learning trials is proposed to be 5 and a minimum of 1 execution per trial. The activation function that was used on the deep layers was Rectified Linear Unit(ReLU). According to state of the art, ReLU is the most used activation function on models.

The following hyperparameter set up was tested on each scenario:

4.1.6.3. Convolutional Filter Tuning

The objective is to find the optimal number of filters in the convolutional layers on the based model. The base model is a ResNet50 pre-trained using the ImageNet dataset. The output of it is a flatten layer. The based model was tuned from 32 to 256 with a step of 32.

4.1.6.4. Dense Layer Units Tuning

The objective is to find the optimal number of dense units and convolutional filter in the model. The number of dense unit hyperparameter was set on the ResNet50 network, where the first deep layer was tuned with a number of units from a range of 32 to 512 with a step of 128. The second deep layer was tuned with a number of units from a range of 32 to 256 with a step of 128. The output layer is a dense layer

with 256 units.

4.1.7. Data Collection

Families in the Wild (FIW) is the database that was used. This is the biggest dataset for kinship image analysis. This database contains 300 families where 1828 are family members and, 10255 are the images in total. The dataset was reduced to 144 families. Data were process before start the testing and use of the model:

1. For each family is required a manual innervation where bad quality photos are going to be removed. Blur pictures, profile faces and use of hats, sunglasses or wigs were considered as delete criteria.
2. Originally, FWI dataset has a file structure where family members are stored in different directories. The file structure was changed to match a simpler structure where all children were stored in a directory call K, mother images in an M directory, father in F, mother from a different family was stored on NM directory and NF for father from a different family.
3. Mother and father from a different family were copied to each family to provide the negative image that is needed for the triplets.
4. Images stored in M, F, NM and NF was balanced to match the same amount of images as the total of kids files. This step prevents to have triplets with no mother nor father and guarantee reliability in the model.

4.1. Experiment Design

5. All pictures was transformed into ages of 5, 15 and 30 years old using the Life Span Age Transformation Synthesis (LATS)[38] technique.
6. The new dataset once is cleaned was stored in a public GitHub account.
7. The pictures was resized to 200x200 pixels using RGB as scale.
8. The 80% of the images was used as the training dataset and the left 20% as the validation dataset.

4.1.8. Analysis

The proposed model was evaluated by using different scenarios that include age transformation and model tuning. The scenarios will run in the dataset . The results of these metrics was shown using graphs and tables for easy comparison. For the proposed model, these was the metrics to use since are the most common metrics used in the state-of-the-art for research:

4.1.8.1. Accuracy

This metric was used to measure the proportion of correctly identified kinship pairs over other evaluated pairs. This metric has been used in models, where it is used to evaluate kinship model performance in different scenarios[30, 35].

4.1.8.2. F1-score

A high F1-score indicates a balanced performance of the model in identifying kinship pairs. Studies demonstrate the importance of F1-score in ensuring robust

4.1. Experiment Design

performance in models, where both false positives and false negatives need to be minimized [29, 32].

4.1.8.3. Mean Squared Error (MSE)

Using MSE helps in ensuring that the model tuning process does not lead to overfitting. It provides a quantitative measure to track the errors during training and testing phases, ensuring that the model generalizes well to unseen data. This metric is particularly useful in tuning processes[1].

4.1.9. Validity

Validity is important in research to determine credibility of the findings. For this work, credibility of all output data from the experiments was assessed in four different dimensions: internal, external, construct and reliability.

4.1.9.1. Internal Validity

To prove and sustain internal validity of obtain data after execute the experiments on the proposed model, a 5-fold cross-validation was performed. 5-fold cross-validation allows comprehensive assessment of the model's capabilities, where each fold serves as a validation set once, and the rest of the folds are the training set[3]. This makes sure that all data points are used both training and validation and obtain reliable estimate of model's performance and helps to reduce overfitting[26, 32]. These were the steps to perform the 5-fold validation on each proposed age transformation scenario(5, 15, 30):

4.1. Experiment Design

1. The selected 144 families was shuffled to ensure randomness.
2. The dataset was divided into five folds. Since there are 144 families, each fold will have 28 families.
3. Each fold was used as the validation dataset, while the rest of the four folds was the testing dataset. Accuracy, F1-score and MSE was recorded for each fold iteration.
4. Finally, an average was computed, taking as input the results of each metric from the five folds to get a final result.

It is important to point out that the families on the database was manually select and requires a manual preprocessing to filter images with poor quality. The database will also be transformed by using the LATS technique[38]. Since hyperparameter tuning was part of the training, it was considered as a possible variant of the results. That is why it was taken into consideration to not cause noise on the final results. The same hyperparameters was used on each fold.

4.1.9.2. External Validity

Due to the scope of this investigation, the evaluation of the proposed model is conducted on database subset. This dataset is being used by different research in area, making it a widely recognized dataset[17]. However, using a subset of this dataset causes a limitation related with the findings which may not be directly generalizable to other kin datasets such as Cornell dataset.

This work aims to approach the age gap challenge in but, limitation of external

4.1. Experiment Design

validity due to data limitation is acknowledged. The findings of this work still offers significant contributions to the kinship field. Future work looks to extend the evaluation to more datasets to validate and generalize the results. The reason behind the use of a subset of 144 families is driven by these considerations:

- Analyze and processing all the families from the Families in the Wild database requires computational resources and time using the proposed scenarios for age transformation and . That is why focusing on a subset allows more manageable data for processing and evaluation.
- Families in the Wild (FIW) database contains poor quality images which are incompatible with the LATS age transformation technique that was used. A manual filter needs to be performed in order to select the best images per family. That is why a manual process filtering will focus on a subset of the database.
- The manual filtering process will consider to include diverse families, including diverse age ranges, amount of images per family member and image quality which will allow robustness on the results.
- By limiting the dataset, the evaluation of the proposed model was conducted in different scenarios. This approach will allow more detailed analysis that will help to identify capabilities and areas of improvement.

4.1.9.3. Construct Validity

To ensure that the proposed model effectively evaluates by integrating age transformation and model tuning methodologies, these are the proposed areas to address:

4.1.9.4. Relation between facial features and kin verification

This work is based in the premise that thought facial features can be analyzed and determine the kinship relationships. This is covered by the theoretical background supported by the state-of-the-art analysis for facial recognition and . Studies has shown that certain facial features are indicative of familiar relationships, demonstrating the effectiveness of discriminative deep learning metric techniques in capture of features for [4, 9].

4.1.9.5. Age Transformation

The approach of using an age transformation techniques to mitigate the variation in facial features due to aging is aligned with the theoretical understanding that facial features change over time and, transforming images to a certain age range can improve verification accuracy[1, 5, 20].

4.1.9.6. Model Tuning

This work aims to use as model tuning methodologies to ensue that the algorithm is optimized for performance according to the proposed scenarios. There is evidence that suggests is important for enhancing the performance of machine learning models

[9, 19, 26].

4.1.9.7. Use of established techniques

The implementation of established techniques like Siamese Networks, Life Span Age Transformation Synthesis (LATS) and ResNet50 architecture ensures that constructs are measured using validated and reliable methods from the related research[1, 24, 29, 32].

4.1.9.8. Evaluation of metrics

The performance of the proposed model was evaluated by using the most popular metrics implemented in research: accuracy, F1-score and Mean Square Error (MSE). These metrics are used and accepted in the machine learning field and provide robust performance measure in verifying kinship relations[4, 5, 9].

4.1.9.9. Reliability

The methods and metrics used in this work to ensure reliability are covered by different testing scenarios. Each scenario aims to ensure quality and integrity of each result by applying multiple validation techniques, such as:

- 5-fold cross-validation, which will help to mitigate the risk of overfeeding by the use of and ensuring that the model's performance is not dependent on a particular set of data thanks to the implementation of folds.
- Implementation of standard metrics like accuracy, F1-score and Mean Squared Error (MSE) to evaluate the proposed model.

The use of folds and recognized metrics will affirm the reliability of the proposed model, ensuring that the final results are trusted, ensuring that the model’s performance is evaluated comprehensively, allowing to capture different aspects of the model’s behavior across various data splits.

4.1.10. Limitations

Families in the Wild (FIW) dataset has several limitations such as unbalanced images and poor quality. Manual intervention is required to clean the data that was used. These are some of the limitations that has been identified:

- Not all images could be transformed, resulting in some individuals having only one image that could not be transformed or a few poorly transformed images unsuitable for kinship relationship analysis.
- The FIW database contains low-quality images, with side poses, dark glasses, patches, etc., which affect the transformation.
- Human intervention is required to clean the database before transformation, which may leave some images that are not good or result in the elimination of high-quality images.
- The database contains many relationships per family. In some families, like F0004, there are grandchildren, which introduces additional mother-child relationships. There are also siblings within a family where one of the siblings is the parent, resulting in in-law relationships.

4.1. Experiment Design

- Some individuals have photos from their infancy, which do not transform well into adult images. Research has shown that transforming images of infants into adult faces is particularly challenging where the shape and appearance of a baby’s face can change drastically by adulthood, making it difficult to model and predict these changes accurately. Real-life photos of children are tough to age-progress due to factors like variable lighting, expressions, and the baby-like characteristics of faces. It’s especially hard to produce accurate images for children younger than age 5, as their facial features closely resemble that of a baby and change significantly as they grow older[39].
- When cleaning the original database, some individuals are left with only one image because the others were of poor quality. In some cases, the database only includes one photo per individual, and it may be of poor quality.

These limitations highlight certain challenges encountered when working with the FIW dataset, such as occasional image quality issues and the need for manual data cleaning. Despite these challenges, the dataset remains a rich resource for kinship verification research.

This chapter details the developed algorithm and mechanisms designed to evaluate the proposed hypothesis. The contents of this section provide a comprehensive description of the methodology, model architecture, and hyperparameter tuning techniques utilized in the experimental process. Each aspect of the model’s design is explained to demonstrate how the model was planned and structure to meet the objectives and prove the hypothesis.

5.1. Proposed algorithm

This section presents the proposed algorithm for kinship verification using deep learning techniques integrated with age transformation. The objective is to create a model that can check if people pictures are kin related or not. The algorithm is based on the use of the Families in the Wild (FIW) dataset, which serves as the base for training and validating the model. The proposed model leverages the ResNet50 architecture as an embedding model in a Siamese Network, with hyperparameter tuning to enhance the model’s performance.

Diagram 5.1 shows a visual representation of the proposed model, where the process begins with the data preparation stage and the FIW dataset is cleaned and organized to suit the needs of the model. This includes managing image quality, age transformations, and organizing the data into appropriate categories. Once data preparation is complete, the next phase involves building the embedding model using ResNet50, which extracts crucial features from the facial images. These features are then processed through the Siamese Network, where the model learns to distinguish

5.1. Proposed algorithm

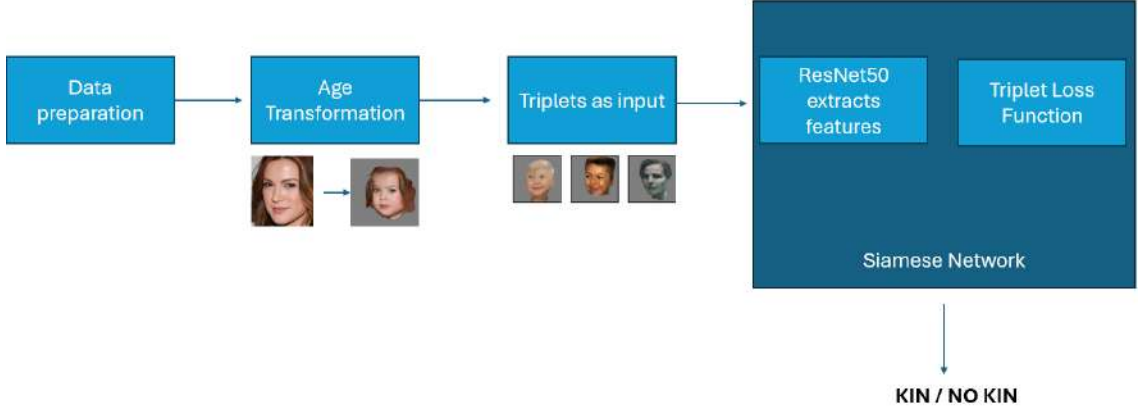


Figure 5.1: Representation of the proposed model.

between familial and non-familial relationships using triplet loss.

A detailed explanation of each step of the algorithm, including data preparation, embedding model construction, and the complete Siamese Network model, is provided in the following subsections.

5.1.1. Data preparation

The Families in the Wild dataset is the one selected to run the proposed model. This dataset contains more than 1000 families but, not all of them are eligible to be used as part of testing and validation for the proposed model. The photos' database contains many low-quality images, such as those with side poses, dark glasses, or patches, all of which affect the transformation. Additionally, the database contains a variety of relationships per family, including complex ones such as grandchildren or in-laws making it out of the scope of relations since the proposal covers mother-children and father-children ones. Some individuals have childhood photos that don't transform well into adult images, and after cleaning, some individuals are left with

5.1. Proposed algorithm

only a single image, often of poor quality. In some cases, the database itself provides only one photo per individual, which may also be of low quality and to run the model we need the major amount of photos per family member. After the manual clean up of the photos, the transformation of each age range happens using the LATS algorithm. The images must then be reorganized to optimize the model's input. The methods developed to handle data preparation are explained below:

`createDirs(familiesDir)`

This method is responsible for creating directories for each family in the dataset. Specifically, it organizes the family into subdirectories for different relationships: "K" (kids), "M" (mother), "F" (father), "NM" (negative mother), and "NF" (negative father). These directories will hold the respective images for further processing.

`organizePhotosInDirs(photosDir, csvMatrix)`

This method processes the images according to relationships defined in a CSV file that comes as part of the Families in the Wild dataset and organizes them into the appropriate directories created by `createDirs`. It filters images that lack an "Age" attribute in their filename and moves valid images to the subdirectories based on family relationships (e.g., "M", "F", "K").

`removeDirs(familiesDir)`

This method iterates through the dataset to identify and remove any empty family directories after the images have been processed. It specifically looks for directories

5.1. Proposed algorithm

matching a pattern (e.g., "***MID***") and deletes them if they are empty.

`createNoMomDad(familiesDir)`

The `createNoMomDad` function deals with the creation of "No Mother" or "No Father" images for families that lack parental images. It duplicates mother and father images from other families into the appropriate directories for families that don't have such images. If the last family in the dataset lacks parental images, the function copies the parents from the first family.

`balanceData(familiesDir)`

The `balanceData` method is used to balance the dataset by ensuring that the number of mother and father images matches the number of kid images in each family. If the number of images in any category is less than the number of kid images, the method copies existing images to match the count. If there are too many images, it randomly removes images to balance the dataset.

Once the images are clean and organized, the dataset to train the model needs to be created based on the subset taken from the Families in the Wild dataset. This dataset has the characteristic that it needs to arrange the images into the input triplets [anchor, positive, negative] using an RGB channel and resize them to 200x200 pixels. To manage this, the methods explained below were created to support the dataset creation process:

5.1. Proposed algorithm

`preprocess_image(filename)`

The `preprocess_image` function is responsible for loading an image, converting it to the appropriate data type, resizing it, and returning the preprocessed image. First, the image is loaded using `tf.io.read_file()`. It is then decoded into a tensor using `tf.image.decode_jpeg()`. The image is converted to a float32 data type using `tf.image.convert_image_dtype()` for compatibility with neural networks. Lastly, the image is resized to the target shape (200x200) using `tf.image.resize()`.

`preprocess_triplets(anchor, positive, negative)`

The `preprocess_triplets` function processes three images corresponding to the anchor, positive, and negative inputs for the Siamese network. It takes the filenames of these images as inputs and uses the `preprocess_image()` function to load and preprocess each image. This function returns a tuple containing the preprocessed anchor, positive, and negative images.

5.1.1.1. Image Loading and Organization into Triplets

The code block for image loading and organization into triplets iterates over family directories, loading images corresponding to different family members such as fathers, mothers, and children. Based on the family member, the images are added to respective lists, including `fatherImages`, `motherImages`, and `kidsImages`. Once all images are loaded, datasets for the anchor, positive, and negative samples are created using `tf.data.Dataset.from_tensor_slices()`. The datasets are then combined using `tf.data.Dataset.zip()` to form a dataset of triplets. Finally, the dataset is

5.1. Proposed algorithm

preprocessed by mapping the `preprocess_triplets()` function, ensuring that each triplet of images is correctly formatted and preprocessed for model input.

5.1.1.2. Dataset Splitting and Batching

After creating the dataset of triplets, the dataset is split into training and validation sets. The training set comprises 80% of the dataset, while the validation set is 20%, and the splitting is done using the functions' `dataset.take()` and `dataset.skip()`. The training and validation datasets are batched with a batch size of 3. To improve data loading performance, prefetching is applied using `train_dataset.prefetch()`.

5.1.2. ResNet50 embedded model

The proposal for the Siamese Neural Network architecture for kinship verification is build on top of a ResNet50 architecture that works as the embedding model in the Siamese Network, which objective is to extract the features. The Siamese network, which computes the distance between anchor, positive, and negative image embeddings. The model uses hyperparameter tuning to fine-tune both convolutional and dense layers on ResNet50, making it adaptable for kinship verification tasks. The custom training loop ensures that the model optimizes the triplet loss, making it effective in distinguishing between related and unrelated individuals. The proposed model is explained in detail in the next subsections.

5.1.2.1. Embedding model

The embedding model plays a crucial role in the kinship verification process, as it is responsible for transforming input images into a feature vector (embedding) that captures the essential characteristics of the image. These feature vectors are numerical representations that encapsulate the unique attributes of an individual's facial features, such as the shape, texture, and distinct patterns that are important for identifying familial relationships. In this research, a ResNet50-based architecture is used as the backbone of the embedding model, which extract high-level features from facial images. Algorithm 1 shows a pseudocode of the structure created to handle the ResNet50 networks in charge of the feature extraction.

Below, subsections explain in detail the pseudocode showed in Algorithm 1.

5.1.2.2. Pre-trained Base Model (ResNet50)

The embedding model uses ResNet50, a deep CNN pre-trained on the ImageNet dataset, as the base model. ResNet50 is chosen for its powerful feature extraction capabilities and is configured with `include_top=False`, which means that the fully connected layers at the top are removed to allow custom layers to be added. The input shape is set to `(200, 200, 3)` to handle RGB images of size 200x200.

5.1.2.3. Additional Convolutional Layers

After the ResNet50 base, the code adds additional tunable convolutional layers:

- `hp_conv_filters` is a tunable hyperparameter that defines the number of filters in each convolutional layer, ranging from 32 to 256 with steps of 32.

5.1. Proposed algorithm

Algorithm 1 Pseudo Code for Embedding Model with Hyperparameter Tuning

Input: Input: Hyperparameters hp , Target Shape $target_shape$

Output: Output: Trained Embedding Model

Step 1: Initialize base CNN using ResNet50 pre-trained on ImageNet

Step 2: Add tunable convolutional layers

2.1: Choose number of convolutional filters from range $[32, 256]$

2.2: Apply 2D with chosen filters and ReLU activation

2.3: Apply 2D

2.4: Add a second 2D layer with the same number of filters

2.5: Apply 2D

Step 3: Flatten the output of the convolutional layers

Step 4: Add the first Dense (fully connected) layer

4.1: Choose the number of units from range $[32, 512]$

4.2: Apply layer with chosen units and ReLU activation

4.3: Apply

Step 5: Add the second Dense layer

5.1: Choose the number of units from range $[32, 256]$

5.2: Apply layer with chosen units and ReLU activation

5.3: Apply

Step 6: Add output layer with 256 units

Step 7: Set trainable layers in the base CNN starting from 'conv5_block1_out'

Step 8: Return the embedding model

5.1. Proposed algorithm

- Each convolutional layer uses a filter size of (3, 3) and the activation function is chosen as `relu` from a list of possible activation functions.
- Two max-pooling layers are added after each convolutional layer to reduce the spatial dimensions of the feature maps.

5.1.2.4. Dense Layers

- `hp_unitsD1` defines the number of neurons in the first Dense layer, ranging from 32 to 512. It uses `relu` as the activation function.
- `hp_unitsD2` is another tunable hyperparameter for the second Dense layer, with values ranging from 32 to 256. It uses `relu` activation.
- Batch normalization is applied after each Dense layer to stabilize and speed up training.

5.1.2.5. Output Layer and Trainable Layers

The final output of the embedding model is a 256-dimensional feature vector. The last few layers of ResNet50, starting from `conv5_block1_out`, are set to be trainable while the rest are frozen. This allows the model to fine-tune the deeper layers while preserving the pre-trained weights in earlier layers.

5.1.3. Siamese Network Model

The Siamese network is created using the anchor, positive, and negative image inputs where and the distances computed by the Distance Layer are the output of the

5.1. Proposed algorithm

Siamese network. The Siamese network model consists of inputs anchor, positive, negative which forms the triplet and outputs the pair of distances [ap_distance, an_distance] for each triplet, this means that for the triplet with index N the results of the distances meaning is kin or not are going to be ap_distance[N] for the result of anchor and positive and an_distance[N] for the result of the anchor and negative. A Siamese Model Class was created to manage different activities inside the model. Some of these activities are build the model with the ResNet50 model which works as the embedded model to extract the features, it also defines the distance layer for computing the embedding differences, it defines the input layers for positive, negative and anchor, it computes the distances between the embedding and creates and compiles the Siamese Network model.

5.1.4. Siamese Model Class

The `Siamese Model` class defines the training loop and the computation of the triplet loss. This loss function helps the model learn to minimize the distance between the anchor-positive pair and maximize the distance between the anchor-negative pair. Algorithm 2 shows the pseudocode of the proposed Siamese model.

5.1. Proposed algorithm

Algorithm 2 Pseudo Code for Siamese Network with ResNet50 Embedding

Input: Hyperparameters hp , Triplet inputs: anchor, positive, negative

Output: Trained Siamese model

Step 1: Build Embedding Model with ResNet50

$embedding \leftarrow build_embedding_model(hp)$

Step 2: Define Distance Layer for Computing Embedding Differences

$ap_distance \leftarrow tf.reduce_sum(tf.square(anchor - positive))$

$an_distance \leftarrow tf.reduce_sum(tf.square(anchor - negative))$

return ($ap_distance, an_distance$)

Step 3: Define Input Layers for Anchor, Positive, and Negative

$anchor_input \leftarrow layers.Input(shape = (target_shape + (3,)))$

$positive_input \leftarrow layers.Input(shape = (target_shape + (3,)))$

$negative_input \leftarrow layers.Input(shape = (target_shape + (3,)))$

Step 4: Compute Distances between Embeddings

$distances \leftarrow ()$

$(embedding(resnet.preprocess_input(anchor_input)),$

$embedding(resnet.preprocess_input(positive_input)),$

$embedding(resnet.preprocess_input(negative_input)))$

Step 5: Create Siamese Network Model

$siamese_network \leftarrow Model(inputs = [anchor_input, positive_input,$
 $negative_input], outputs = distances)$

Step 6: Define the Siamese Model Class

$siamese_network, margin=0.5$

 Initialize loss tracker and accuracy tracker

$train_step(data)$

 Start gradient tape

$loss \leftarrow$ Compute triplet loss from data

$accuracy \leftarrow$ Compute accuracy from data

 Apply gradients to model weights

 Update loss and accuracy metrics

return tracked metrics

Step 7: Compile the Siamese Model

$siamese_model \leftarrow SiameseModel(siamese_network)$

$siamese_model \leftarrow siamese_model.compile(optimizer =$
 $Adam(0.0001), metrics[accuracy])$

Step 8: Return the Compiled Siamese Model

return $siamese_model$

5.1. Proposed algorithm

The **SiameseModel** class defines the main architecture for the Siamese network used in the proposed kinship verification model. This class inherits from the Keras **Model** class, which allows custom behavior for training and testing loops. The primary objective of this class is to compute the **triplet loss** based on the embeddings of anchor, positive, and negative images, and to optimize the model using this loss.

The class is initialized with two primary components: the **siamese_network** and the **margin**. The **siamese_network** represents the core model responsible for generating the embeddings of the anchor, positive, and negative samples, and for computing the distances between the anchor-positive and anchor-negative pairs. The **margin** is a hyperparameter used in the triplet loss function to define how much closer the anchor-positive pair should be compared to the anchor-negative pair.

The key method in the class is **train_step**. In this method, the triplet loss is computed during the training process. A **GradientTape** is used to record the operations and compute the loss. The gradients of the loss with respect to the model's weights are then computed. These gradients are applied to update the model's parameters using the specified optimizer.

The **test_step** method is similar to the **train_step**, but it is used during the evaluation phase on validation or test data. This function computes the loss and accuracy without updating the model weights.

The **_compute_loss** method calculates the triplet loss. This loss is defined as the difference between the anchor-positive distance and the anchor-negative distance, enforced by the margin parameter. The result is passed through the **tf.maximum**

5.1. Proposed algorithm

function to ensure non-negative values, thereby preventing the loss from going below zero.

The `_compute_accuracy` method computes the accuracy by comparing the anchor-positive distance with the anchor-negative distance. If the anchor-positive distance is smaller than the anchor-negative distance, the prediction is considered correct.

Finally, the `metrics` property returns the loss and accuracy trackers, which keep track of the average values for the loss and accuracy during training or testing.

The **SiameseModel** is compiled with the `Adam` optimizer and the model is set to track accuracy as an evaluation metric.

5.1.4.1. Distance Layer

The `DistanceLayer` is a custom layer that computes the Euclidean distance between the embeddings of the anchor-positive and anchor-negative image pairs:

- It calculates the squared difference between the anchor and positive embeddings, and between the anchor and negative embeddings.
- The distances (`ap_distance` and `an_distance`) are the output of the Siamese network.

5.1.4.2. Loss Function

The triplet loss is defined as:

$$L(A, P, N) = \max(\|f(A) - f(P)\|^2 - \|f(A) - f(N)\|^2 + \text{margin}, 0) \quad (5.1)$$

where $f(A)$, $f(P)$, and $f(N)$ are the embeddings of the anchor, positive, and negative images, respectively, and `margin` is a hyperparameter that defines how much larger the distance between the anchor-negative pair should be compared to the anchor-positive pair.

5.2. Model Compilation

The `SiameseModel` class is compiled with the Adam optimizer and accuracy as the evaluation metric. The model is trained with a custom training loop, which applies the triplet loss to optimize the Siamese network. It also calls the hyperparameter tuning while training on folds and after the model is trained and when model predict is called, the F1-Score method is used to get the values for each fold. The next sections explain in detail the used tuning and the F1-Score function implemented.

5.2.1. Hyperparameter Tuning

The hyperparameter tuning process in the proposed model is managed through the function that creates the ResNet50 where some hyperparameters from the dense layers are tuned using Keras Tuner. This allows for an automated search to find the best set of hyperparameters for the model. First, the number of filters in the convolutional layers is a tunable hyperparameter. This is defined with `hp.Int('conv_filters', min_value=32, max_value=256, step=32)`. It searches for the optimal number of filters (within the range of 32 to 256) to apply to the convolutional layers, allowing flexibility in learning capacity. Next, the

5.2. Model Compilation

activation function for the convolutional layers is also tunable, with `relu` being the available option. This is specified with `hp.Choice('conv_activation', values=['relu'])`. The goal here is to choose the optimal activation function that will best help the model learn non-linear relationships in the data. Additionally, the number of neurons in the dense layers is a tunable hyperparameter. For the first dense layer, `hp.Int('unitsD1', min_value=32, max_value=512, step=128)` allows for a range of 32 to 512 units. For the second dense layer, `hp.Int('unitsD2', min_value=32, max_value=256, step=128)` defines the search space from 32 to 256 units. This flexibility allows the model to search for the best configuration in terms of complexity and capacity in the dense layers. To stabilize and speed up the training process, `Batch Normalization` is applied after each dense layer. This helps improve convergence and ensures that the model generalizes better by normalizing the activations. The hyperparameter optimization is carried out using the `Random Search` strategy in Keras Tuner. During training, the model is built 3 times with different hyperparameter combinations, and performance (measured by validation accuracy) is evaluated to find the best-performing set of hyperparameters. Each of these 3 times the model is trained during 15 epochs. This is the Python code used to start the tuning search using the `RandomSearch` method provided by Keras Tuner:

Listing 5.1: Model Tuning for the Current Fold

```
# Define the model tuning for the current fold
tuner = RandomSearch(
    build_siamese_model, # Assuming build_siamese_model is defined elsewhere
    objective='val_accuracy',
    max_trials=3, # Adjust based on your needs
    executions_per_trial=1,
```

5.2. Model Compilation

```
|     directory=f'my_dir_fold_{fold}',  
|     project_name=f'siamese_network_fold_{fold}',  
| )  
| # Perform the hyperparameter search on the current fold  
| tuner.search(train_fold, epochs=15, validation_data=val_fold, verbose=1)
```

Finally, once the optimal hyperparameters are determined, the embedding model is integrated into the Siamese network. This ensures that the chosen hyperparameters contribute to effective and accurate kinship verification in the Siamese network model.

5.2.2. F1-Score method

The F1-Score is an essential metric in evaluating the performance model network. The proposed model used it for the kinship verification tasks as a manner to check the models' behavior. The implemented F1-Score method evaluates the model's ability to distinguish between related (anchor-positive) and unrelated (anchor-negative) pairs based on their distances. The function `evaluateModel` takes three main inputs: the trained Siamese model, a dataset of triplets consisting of anchor, positive, and negative images, and a threshold value used to classify whether pairs of images are related or not which is set to 0.5. The model processes the input triplets by predicting two distance values: the anchor-positive distance (`ap_distance`) and the anchor-negative distance (`an_distance`). These distances represent how similar the anchor image is to the positive image and how dissimilar the anchor is from the negative image, respectively.

For each triplet, the function compares the anchor-positive distance with the threshold. If the distance is smaller than the threshold, the model predicts that the

images are related, and the predicted label is 1. But, if the anchor-negative distance is greater than the threshold, the model predicts that the images are unrelated, and the predicted label is 0. The ground truth values are set as 1 for the anchor-positive pairs (since they are related) and 0 for the anchor-negative pairs (since they are unrelated). Once the true and predicted labels have been generated for the entire dataset, the F1-Score is computed. Algorithm 3 shows a pseudocode of the implemented method.

The F1-Score is a measure that balances both precision and recall, making it particularly useful in scenarios such as kinship verification, where both false positives and false negatives must be minimized. The F1-Score is calculated using the formula:

$$F1 = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$$

Precision is the fraction of true positive predictions out of all positive predictions made by the model, while recall is the fraction of actual positive pairs correctly identified by the model. The final F1-Score provides a comprehensive evaluation of the model's performance across each fold in the cross-validation process.

5.2. Model Compilation

Algorithm 3 Pseudo Code for F1-Score applied to Siamese Network

Input: Dataset of triplets (anchor, positive, negative), trained Siamese model, threshold

Output: F1-Score for each fold

Function `evaluateModel(model, dataset, threshold):`

 Initialize empty lists: `y_true` and `y_pred`

for *batch* **in** *dataset* **do**

 Extract **anchor**, **positive**, and **negative** images from the batch

 (*ap_distance*, *an_distance*) $\leftarrow$ *model.predict*([**anchor**, **positive**, **negative**])

for *i* **in** *range(len(ap_distance))* **do**

 Append 1 to `y_true` for positive pair (**anchor-positive**)

if *ap_distance[i]* < *threshold* **then**

 Append 1 to `y_pred`

end

else

 Append 0 to `y_pred`

end

 Append 0 to `y_true` for negative pair (**anchor-negative**)

if *an_distance[i]* > *threshold* **then**

 Append 1 to `y_pred` kinship

end

end

 Append 0 to `y_pred`

end

 Compute the F1-Score using `y_true` and `y_pred`

return *F1-Score*

The pseudocode of Algorithm 3 enables the calculation of the F1-Score for each fold during cross-validation, allowing for a detailed assessment of the model's ability to generalize to unseen data. By measuring the model's performance in distinguishing between related and unrelated pairs, the F1-Score provides a valuable metric for evaluating the effectiveness of the Siamese network in kinship verification tasks.

5.3. Limitations and Requirements

The kinship verification model developed for the Families in the Wild (FIW) dataset faces several key limitations, primarily stemming from the quality and balance of the dataset itself. The FIW dataset is highly unbalanced, with significant variation in the number of images available for each family and individual. Additionally, many images in the dataset are of poor quality, including issues like side poses, dark glasses, and patches that obscure key facial features. These factors make it challenging to apply transformation techniques effectively across the dataset, limiting the model’s capacity to consistently analyze kinship relationships.

Another significant challenge is the need for manual intervention during data cleaning. Given the inconsistent quality of the images, the data preparation process requires extensive human oversight to filter out low-quality images and ensure that only suitable images are used for training and validation. This manual curation, however, can sometimes lead to the exclusion of high-quality images, and in some cases, unsuitable images may still be retained due to the complexity of the dataset. Moreover, the presence of multiple types of relationships within a single family, such as grandparents, in-laws, and siblings, introduces further complexity, as the model must account for these varying relational dynamics.

The transformation of images, particularly from infancy to adulthood, poses additional difficulties. Some individuals in the dataset may have only a single image, and others may have images from infancy that do not transform well into adult representations, resulting in a lack of suitable data for certain subjects. In

5.3. Limitations and Requirements

such cases, the model may struggle to generalize effectively. These limitations highlight the need for a robust approach that can handle the dataset’s variability and maintain performance despite the challenges associated with poor-quality images and diverse familial relationships.

The requirements for the kinship verification model using the Families in the Wild (FIW) dataset are centered around the need for high-quality, balanced image data and effective preprocessing techniques. The model requires a substantial amount of images per individual to accurately learn kinship relationships, as well as the ability to handle varying facial features across age transformations. Preprocessing is critical to remove noise and artifacts from the images, ensuring that facial features are clearly visible for analysis. Additionally, the model necessitates computational resources capable of efficiently training deep convolutional networks, such as GPUs, given the complexity of the ResNet50 architecture and the need for hyperparameter tuning through methods like RandomSearch. Lastly, the dataset must be cleaned to eliminate low-quality or irrelevant images, ensuring the model has the most suitable data for kinship verification tasks.

Some technical requirements where the model was run were:

1. Personal computer with Linux or Windows operating system with access to internet.
2. A Google Colab Pro account with 215.74 compute units available, a T4 High RAM instance running Python3.
3. Open source Frameworks for deep learning, such as TensorFlow Keras and Life

Span Age Transformation Synthesis (LATS).

4. A GitHub account to store the data and trained models.

5.4. Implementation

The implementation of the proposed kinship verification model is available in the GitHub repository <https://github.com/ppiedra/FIWDatabase.git>. Clone this repository to have access to all the necessary code and datasets used in this research to train the Siamese neural network model, and evaluate its performance. Appendix section 9 has information of the required branch and what it contains. The URL to the repository and branch <https://github.com/ppiedra/FIWDatabase/tree/final>. The model takes 1 hour to run on each transformed dataset based on the proposed evaluation set up, more information about the evaluation mode can be found in 6.

This chapter presents the results obtained from evaluating the proposed kinship verification model across different age transformation scenarios. The evaluation was conducted using a 5-fold cross-validation strategy applied on the transformed dataset where relations father-children and mother-children were tested across age transformation scenarios of 5, 15, and 30 years. Each scenario provides insights into how well the model adapts to variations in age transformations and familial features. The findings and observations from this chapter are further discussed and analyzed in the Discussion chapter.

6.1. Kinship Verification Model Evaluation

To ensure the robustness and reliability of the proposed model, a comprehensive evaluation was conducted using a 5-fold cross-validation strategy. This approach is chosen to enhance the internal validity of the results obtained from the experiments. By employing 5-fold cross-validation, the entire dataset is partitioned into five distinct subsets or "folds," with each fold serving as a validation set once while the remaining folds form the training set. This procedure allows every data point to be utilized for both training and validation purposes, thereby providing a more reliable estimate of the model's performance and helping to mitigate overfitting risks [3, 26, 32].

The evaluation is performed across different age transformation scenarios (5, 15, 30 years) applied for the father-children and mother-children relations. Diagram 6.1 captures the steps part of the evaluation. More detail of each step is explained below:

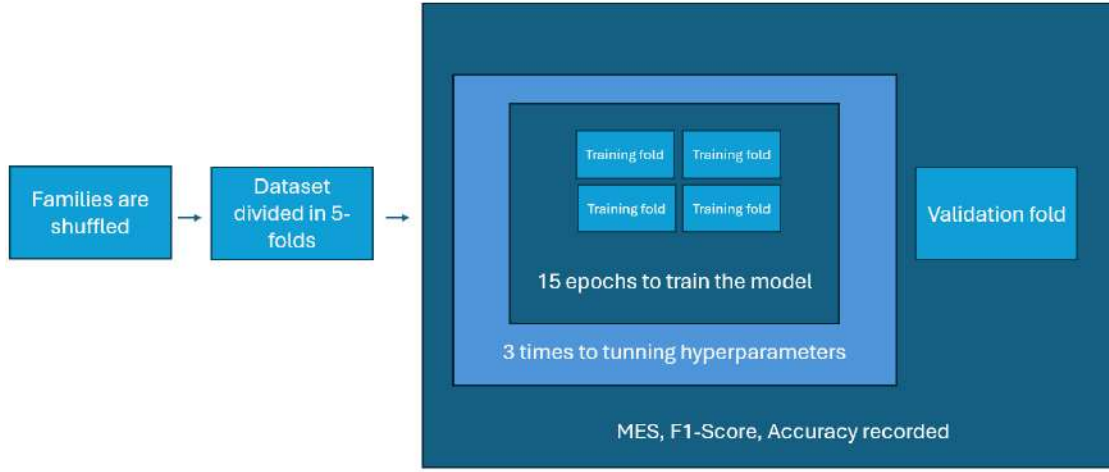


Figure 6.1: Evaluation process of proposed model.

- The selected 144 families from the dataset will be shuffled to ensure randomness and avoid any bias.
- The dataset was divided into five folds, with each fold containing around 28 families. This division ensures that the model is trained and validated on different subsets of data in each iteration.
- In each fold iteration, one fold served as the validation set, while the other four folds were used as the training set. The performance metrics, including Accuracy, F1-score, and Mean Squared Error (MSE), were recorded for each fold iteration.
- After all five iterations, the results of each metric from the folds were averaged to obtain a final performance estimate for the model.

6.1. Kinship Verification Model Evaluation

- Each fold runs on 15 epochs to train the model.
- Each fold tries the hyperparameter adjustment 3 times.
- To complete the analysis, all five folds were evaluated based on the metrics obtained, including accuracy, F1-score, and MSE. The fold with the best performance and the most effective hyperparameter combination was selected.

The LATS technique was employed to transform the database, contributing to the age transformation scenarios. Then, the hyperparameters that were tuned are:

- **conv_filters**

This indicates the number of convolutional filters used in the convolutional layers of the neural network. Convolutional filters are responsible for detecting features in the input images, such as edges and textures.

- **conv_activation: 'relu'**

This specifies the activation function used in the convolutional layers. 'relu' stands for Rectified Linear Unit, which is a common activation function in neural networks. The ReLU function outputs the input directly if it is positive; otherwise, it outputs zero. This helps the model learn complex patterns while maintaining computational efficiency and avoiding issues like vanishing gradients. It was selected as a default value for the ResNet50 model.

- **unitsD1**

This denotes the number of neurons in the first dense (fully connected) layer

of the neural network. Dense layers are used to combine features extracted by convolutional layers and perform the final classification or regression tasks.

- **unitsD2**

This indicates the number of neurons in the second dense layer of the ResNet50 network. Like the first dense layer, this layer also contributes to processing the extracted features before the output layer.

These are the results categorized by relationship type (father-children and mother-children) and the different age transformation scenarios (5, 15, and 30 years). Each scenario provides insights into how the proposed model performs under different age transformations and hyperparameter tuning.

6.1.1. Kinship relation father-children

The analysis results for the relation father-children applied into the proposed kinship verification model provides insights of how the model relates the images by using the face features transformed into different ages. By applying the model to images with varying age transformations (5, 15, and 30 years), it aims to understand how well the model can generalize and maintain accuracy when faced with the natural aging process.

6.1.1.1. 5 years Image Transformation Scenario

For the 5 years image transformation, the best performance happened in fold four, with the next results:

6.1. Kinship Verification Model Evaluation

- It presents an accuracy of 0.85.
- The hyperparameter tuning for the convolutional filters used in the ResNet50 network shows that the model uses 96 filters.
- For the value of neurons selected for the first dense layer (unitsD1) the model was tuned with 160 neurons.
- The number of neurons selected on the second dense layer (unitsD2) the model was tuned with 160 neurons as well.

Table 6.1 shows the hyperparameter tuning folds values used during the training of the proposed kinship verification model. The columns show the values of convolutional filters, the activation function used in the convolutional layers and the number of neurons in the first and second dense layer of ResNet50 network. The convolutional filters on fold 2 were the lowest with 64 while the highest was used in fold 5 with 224.

Table 6.1: Hyperparameter values for 5 years old transformation

Fold	conv_filters	conv_activation	unitsD1	unitsD2
1	128	relu	288	32
2	64	relu	288	160
3	128	relu	288	32
4	96	relu	160	160
5	224	relu	32	32

Figure 6.2 shows the accuracy obtained on each fold. The lowest achieved accuracy happened in fold 3 where the model got a 0.75 meaning the hyperparameter combination might affect the ability to generalize kinship

6.1. Kinship Verification Model Evaluation

verification while 0.85 was the highest accuracy shown in fold 4 where the model successfully learned the relationship between age-transformed image and face kin features.

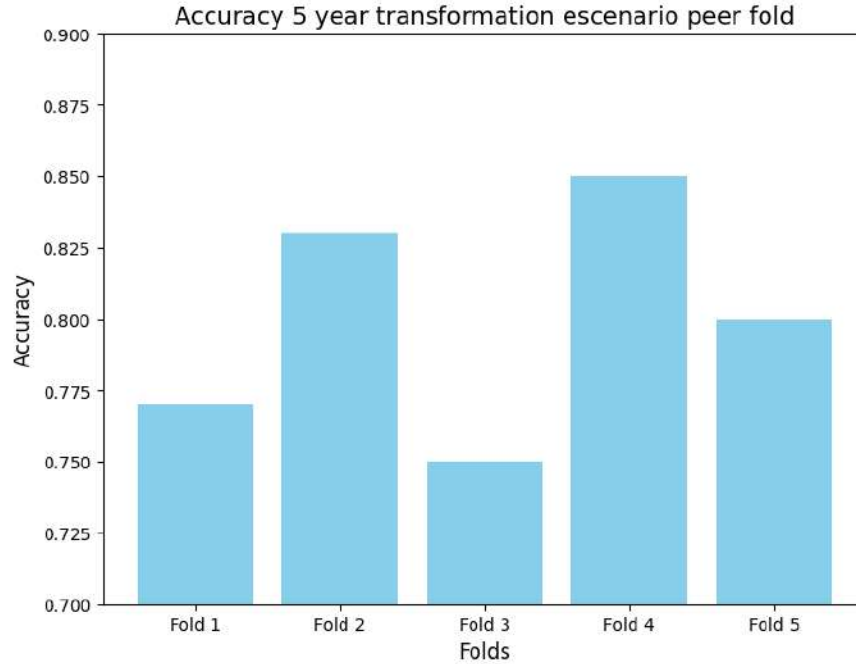


Figure 6.2: Hyperparameter values for each fold in the father-children 5-year transformation scenario.

Table 6.2: Average validation metrics across all folds

Metric	Average Value
Average Validation F1-Score	0.65
Average Validation MSE	0.37
Average Validation Accuracy	0.69

Table 6.2 shows the average validation metrics for the model’s performance across all the five folds, where F1-Score average was 0.65 indicating the balance between precision and recall, 0.37 was the average MES showing the prediction error where

lower values near 0 means better performance and the validation accuracy of 0.69 indicating the average of folds to identify correctly kinship relationships.

6.1.1.2. 15 years Image Transformation Scenario

For the 15-year image transformation, the best fold was the number four, with the following results:

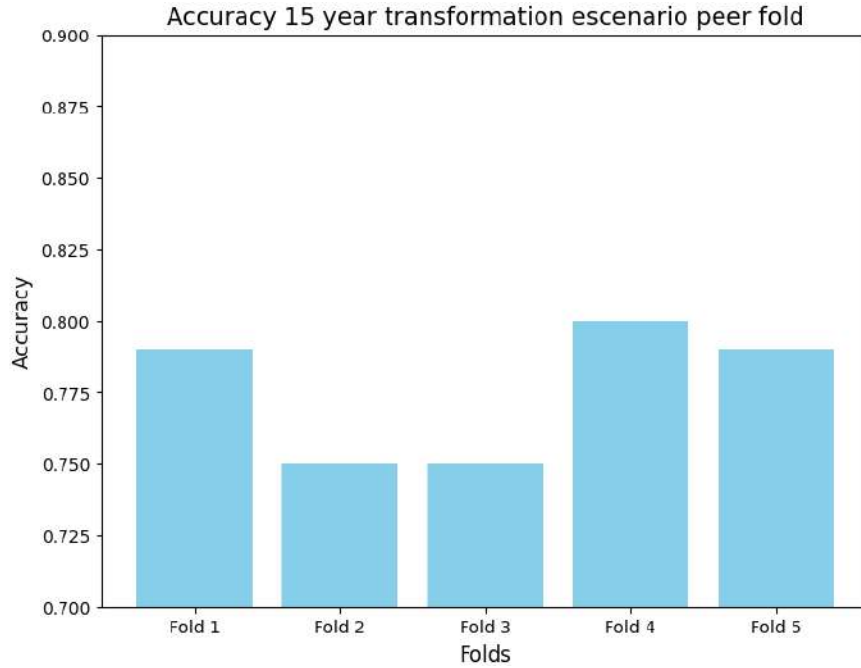
- The model achieved an accuracy of 0.80.
- The hyperparameter tuning for the convolutional filters used in the ResNet50 network shows that the model uses 256 filters.
- For the first dense layer (unitsD1), the model was tuned with 288 neurons.
- The second dense layer (unitsD2) utilized 32 neurons.

Table 6.3 summarizes the hyperparameter tuning folds values used during the training of the proposed kinship verification model when images were transformed to 15 years old. The columns show the values of convolutional filters, the activation function used in the convolutional layers and the number of neurons in the first and second dense layer of ResNet50 network. The convolutional filters on fold 2 were the lowest with 32 while the highest was used in fold 4 and 5 with 256.

Figure 6.3 shows the accuracy obtained for each fold in the 15-year transformation scenario. The lowest accuracy was observed in Fold 2 and 3, where the model achieved both 0.75. The highest accuracy occurred in Fold 4, where the model reached about 0.80. In table 6.4 presents the average validation metrics across all folds for the 15-

Table 6.3: Hyperparameter values for each fold

Fold	conv_filters	conv_activation	unitsD1	unitsD2
1	64	relu	288	32
2	32	relu	288	160
3	96	relu	160	160
4	256	relu	288	32
5	256	relu	416	32

**Figure 6.3:** Hyperparameter values for each fold in the father-children 15-year transformation scenario.

year age transformation scenario. The average validation F1-Score is 0.65, indicating a balance between precision and recall. The average validation Mean Squared Error (MSE) is 0.39, reflecting the model's prediction average error in all folds. The average validation accuracy stands at 0.68, meaning the model correctly predicts kinship relationships in 68% of cases through all the folds in average.

Table 6.4: Average validation metrics across all folds

Metric	Average Value
Average Validation F1-Score	0.65
Average Validation MSE	0.39
Average Validation Accuracy	0.68

6.1.1.3. 30 years Image Transformation Scenario

For the 30-year image transformation, the best fold was the number 4, with the following results:

- The model achieved an accuracy of 0.82.
- The hyperparameter tuning for the convolutional filters used in the ResNet50 network shows that the model uses 64 filters, enabling it to capture a wide range of patterns and details from the input data.
- For the first dense layer (unitsD1), the model was tuned with 288 neurons, allowing it to capture a broad spectrum of feature combinations for more detailed decision-making.
- The second dense layer (unitsD2) utilized 160 neurons. This configuration indicates that the model is designed to be deep enough to handle complex tasks by learning multiple levels of abstraction from the input data.

Table 6.5 summarizes the hyperparameter tuning folds values used during the training of the proposed kinship verification model when images were transformed to 30 years old. The columns show the values of convolutional filters, the activation

6.1. Kinship Verification Model Evaluation

function used in the convolutional layers and the number of neurons in the first and second dense layer of ResNet50 network.

Table 6.5: Hyperparameter values for each fold

Fold	conv_filters	conv_activation	unitsD1	unitsD2
1	96	relu	160	32
2	64	relu	288	32
3	128	relu	288	160
4	64	relu	288	160
5	256	relu	288	32

The results of the accuracy for each fold are show in figure 6.4:

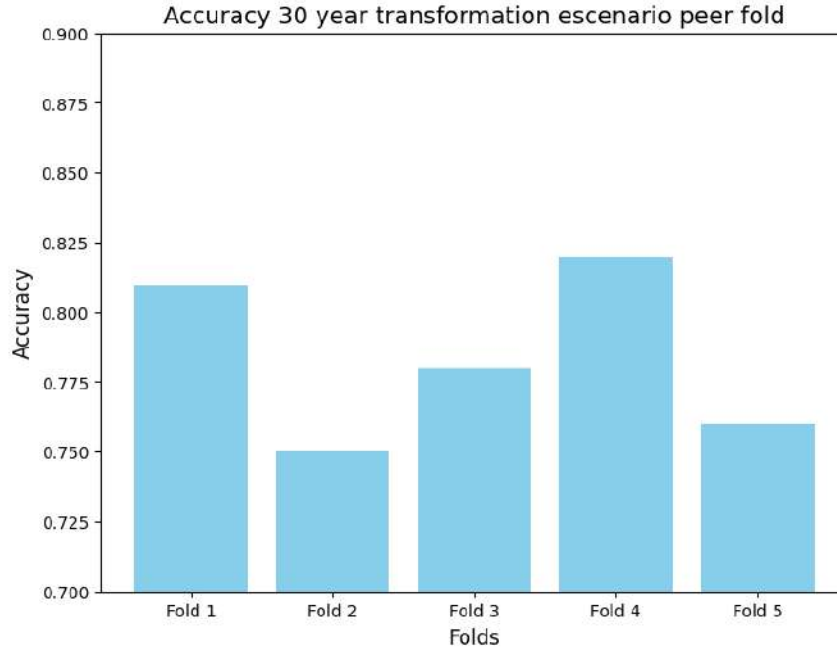


Figure 6.4: Hyperparameter values for each fold in the father-children 30-year transformation scenario.

The average values for the metrics F1-Score, MSE, and accuracy are shown in table 6.6:

Table 6.6: Average validation metrics across all folds

Metric	Average Value
Average Validation F1-Score	0.71
Average Validation MSE	0.35
Average Validation Accuracy	0.80

Fold 4 achieved the highest scores in accuracy, also it showed a low MES indicating that the model effectively learned significant age transformation patterns without overfitting. Overall, the model performed well with 30-year transformations, showing noticeable improvements in both training and validation accuracy over the epochs.

6.1.2. Kinship relation mother-children

The results of the mother-children relationship in the kinship verification model shows how facial features change across different age transformations. The model was tested using images transformed by 5, 15, and 30 years, applying the LATS algorithm to transform the images. The goal was to evaluate how well the proposed model could identify kinship when facial features change over time due to aging.

6.1.2.1. Mother-Children 5 Years Image Transformation Scenario

For the 5-year age transformation applied to the mother-children relationship, the model’s best performance was achieved in fold four, with the following key results:

- The highest accuracy achieved was 0.87.
- The ResNet50 network was tuned to utilize 160 convolutional filters, enabling the model to capture various facial features and details from the images,

facilitating better kinship identification.

- The first dense layer (unitsD1) was configured with 160 neurons, providing the model with the capacity to process a wide range of feature combinations, which might enhance its decision-making capabilities.
- The second dense layer (unitsD2) used 160 neurons, which added depth to the model, helping it to handle more complex tasks for the kinship verification task.

Table 6.7 presents the hyperparameters used for the mother-children 5-year transformation scenario. The folds displayed varying values for convolutional filters, activation functions, and the number of neurons across both dense layers of the ResNet50 network. Such variations across folds demonstrate that the model captures different levels of features, with more filters detecting complex patterns and fewer filters identifying simpler ones. The ReLU activation function remains consistent across folds, providing non-linearity and computational efficiency. The number of neurons in the dense layers also varies, reflecting adjustments in the model’s capacity to process feature combinations.

Table 6.7: Results of hyperparameter values for each fold 5 years old mother-children kinship verification

Fold	conv_filters	conv_activation	unitsD1	unitsD2
1	64	relu	288	160
2	160	relu	416	160
3	160	relu	32	160
4	160	relu	160	160
5	224	relu	288	32

6.1. Kinship Verification Model Evaluation

Figure 6.5 illustrates the accuracy obtained in each fold for the 5-year transformation scenario. The lowest accuracy was recorded in fold 3 with a value of 0.78, suggesting that certain hyperparameter combinations may affect the model’s ability to generalize well for kinship verification. Fold 4 achieved the highest accuracy of 0.87, which means that in this point the combination of hyperparameters made the model improve the facial feature relation between mother and its children.

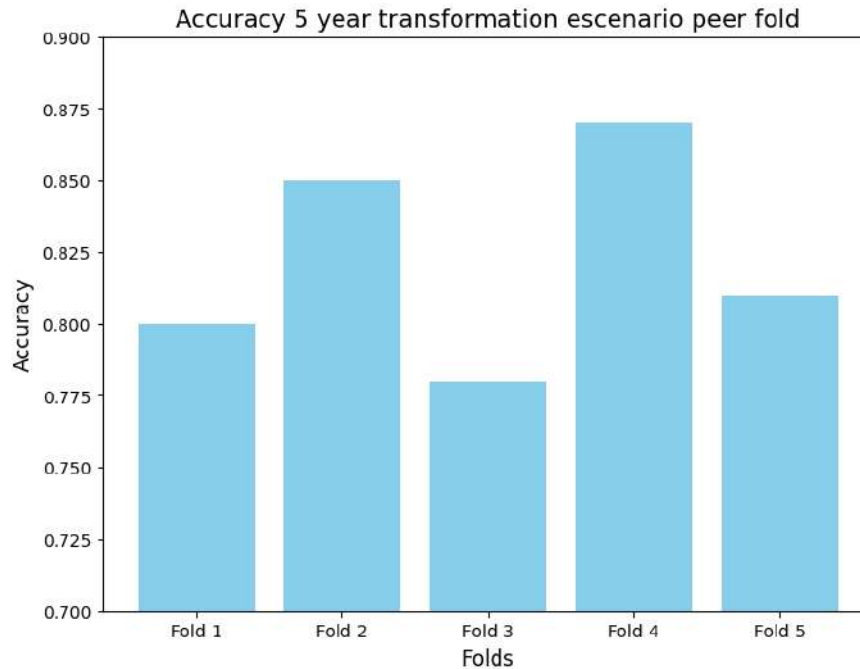


Figure 6.5: Accuracy across folds for 5-year age transformation in the mother-children relationship.

Table 6.8 presents the average validation metrics for the 5-year age transformation scenario. The model’s average validation F1-Score was 0.60. The Mean Squared Error (MSE) averaged 0.36, indicating the model’s low prediction error, and the

validation accuracy stood at 0.70, demonstrating consistent performance in correctly identifying mother-child kinship relationships.

Table 6.8: Average validation metrics across all folds for 5-year mother-children age transformation scenario.

Metric	Average Value
Average Validation F1-Score	0.60
Average Validation MSE	0.36
Average Validation Accuracy	0.70

6.1.2.2. Mother-Children 15 Years Image Transformation Scenario

For the 15-year age transformation applied to the mother-children relationship, the model’s best performance was achieved in fold four, with the following key results:

- The highest accuracy achieved was 0.89.
- The ResNet50 network was tuned to utilize 192 convolutional filters, enabling the model to capture a wide range of patterns and details from the images.
- The first dense layer (unitsD1) was configured with 416 neurons.
- The second dense layer (unitsD2) was configured with 32 neurons.

Table 6.9 provides the hyperparameter tuning values for each fold in the mother-children 15-year transformation scenario. The values of convolutional filters, the activation function used in the convolutional layers, and the number of neurons in the first and second dense layers of the ResNet50 network are displayed. All folds have different hyperparameter combinations, which makes a slight variation in the accuracy results.

6.1. Kinship Verification Model Evaluation

Table 6.9: Results of hyperparameter values for each fold in the mother-children 15-year transformation scenario.

Fold	conv_filters	conv_activation	unitsD1	unitsD2
1	32	relu	288	32
2	192	relu	416	32
3	128	relu	160	32
4	224	relu	416	160
5	128	relu	228	32

Figure 6.6 presents the accuracy obtained across all folds in the 15-year transformation scenario. The lowest accuracy was achieved in folds 4 and 1, where the model reached 0.83 and 0.84. The highest accuracy was achieved in fold 2 with a value of 0.89.

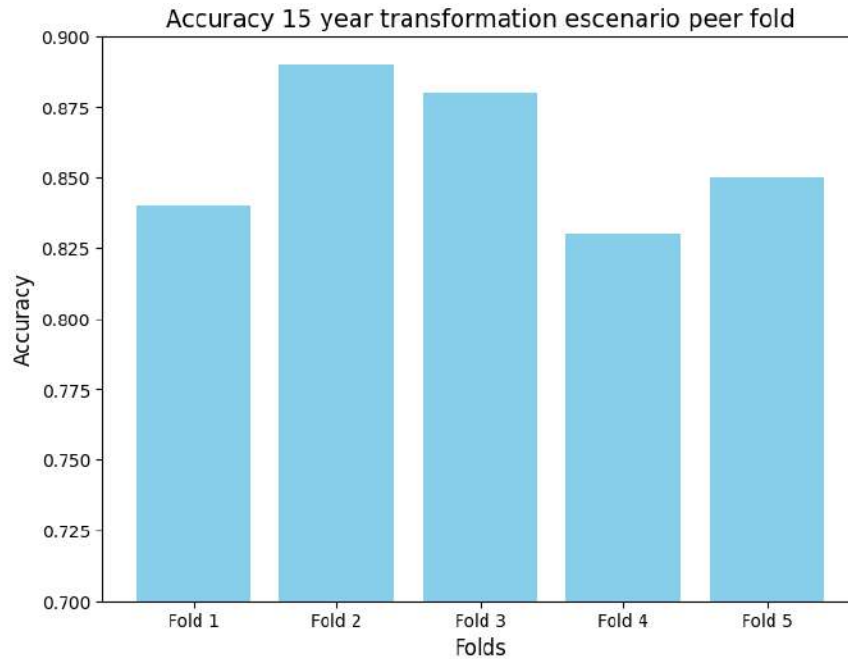


Figure 6.6: Accuracy across folds for 15-year age transformation in the mother-children relationship.

Table 6.10 provides the average validation metrics across all folds for the 15-year transformation scenario. The average validation F1-Score is 0.65, indicating the model’s ability to balance precision and recall across the dataset. The Mean Squared Error (MSE) averaged 0.35, representing the model’s prediction error, while the average validation accuracy was 0.77, indicating that the model correctly predicted kinship relationships in approximately 77% of cases.

Table 6.10: Average validation metrics across all folds for 15-year mother-children age transformation scenario.

Metric	Average Value
Average Validation F1-Score	0.65
Average Validation MSE	0.35
Average Validation Accuracy	0.77

6.1.2.3. Mother-Children 30 Years Image Transformation Scenario

For the 30-year age transformation applied to the mother-children relationship, the model’s performance results are as follows:

- The highest accuracy achieved was 0.88.
- The ResNet50 network was tuned with 96 convolutional filters, allowing the model to extract significant features from the transformed images.
- The first dense layer (unitsD1) was configured with 416 neurons.
- The second dense layer (unitsD2) was configured with 32 neurons.

Table 6.11 presents the hyperparameter tuning values for each fold in the mother-children 30-year transformation scenario. Each fold had different combinations of

6.1. Kinship Verification Model Evaluation

convolutional filters and neurons in the dense layers, contributing to slight variations in accuracy results.

Table 6.11: Hyperparameter values for each fold in the mother-children 30-year transformation scenario.

Fold	conv_filters	conv_activation	unitsD1	unitsD2
1	96	relu	416	32
2	64	relu	32	32
3	224	relu	288	32
4	192	relu	32	160
5	64	relu	288	160

Figure 6.7 presents the accuracy obtained across all folds in the 30-year transformation scenario. The highest accuracy was recorded in fold 1, with a value of 0.88, while fold 5 had the lowest accuracy at 0.82.

Table 6.12 provides the average validation metrics across all folds for the 30-year transformation scenario. The average validation F1-Score was 0.66. The Mean Squared Error (MSE) averaged 0.33, representing the model’s prediction error, while the average validation accuracy was 0.78, indicating that the model correctly predicted kinship relationships in approximately 78% of cases.

Table 6.12: Average validation metrics across all folds for the 30-year mother-children age transformation scenario.

Metric	Average Value
Average Validation F1-Score	0.66
Average Validation MSE	0.33
Average Validation Accuracy	0.78

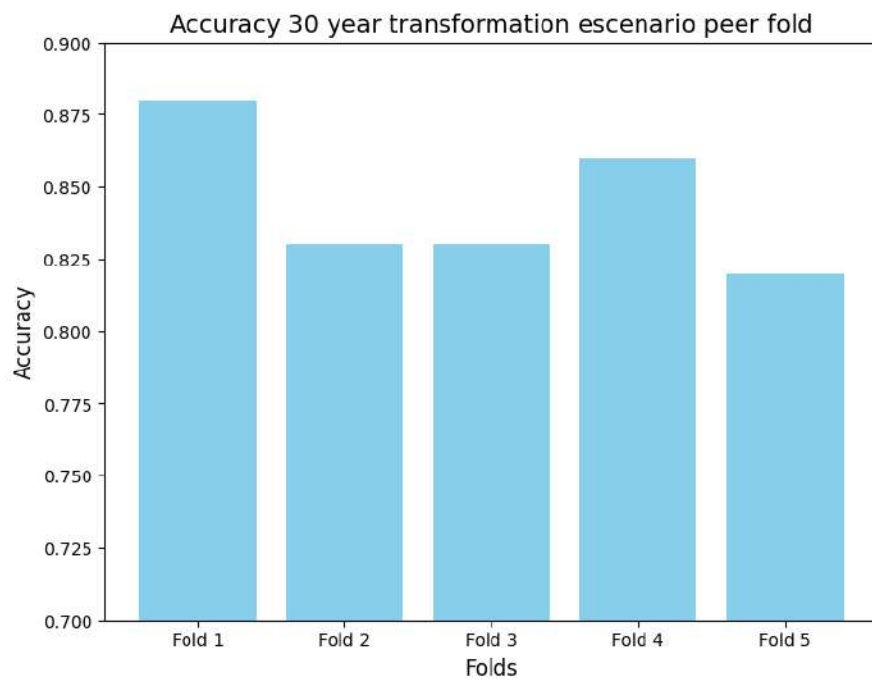


Figure 6.7: Accuracy across folds for the 30-year age transformation in the mother-children relationship.

Chapter 7

Discussion

This section aims to provide a comprehensive analysis of the results shown in chapter Results. The obtained information from the kinship verification models across various age transformation scenarios examines the metrics observed during the training and validation phases. The results show how different age transformations, relationship types, and folds contributed to obtain results to discuss the model’s performance compared with the state-of-the-art.

7.0.1. Father-Children kinship verification analysis

The analysis of the kinship verification model for the father-children relationship across different age transformation scenarios (5, 15, and 30 years) provides insights into the model’s ability to generalize and maintain accuracy when faced with varying degrees of facial aging. In the 5-year image transformation scenario, the model achieved its highest accuracy of 0.85 in Fold 4, using 96 convolutional filters and 160 neurons in both dense layers (unitsD1 and unitsD2). This configuration allowed the model to capture changes in facial features associated with a 5-year age difference effectively, suggesting that the model is able to identifying kinship when the images have younger age differences. 5, 15, and 30 years age ranges were chosen based on medical evidence suggesting that facial features between parents and their children tend to be similar when both are in the same age range. Using photographs of children at one, ten, and twenty years of age and their parents, Christenfeld & Hill [40] reported that people can match fathers to 1-year-olds, consistent with the idea that babies resemble their fathers. It is also proved that childish facial features are more likely to identify kinship and this can be confirmed with the test that was run

where it was found that between transforming images in age ranges of 5, 15 and 30 the kinship verification results better when images are transformed to 5 years old showing that childish facial features are a good technique when researching on kin relations. The consistent performance metrics in this scenario highlight the model's capacity to generalize across the dataset despite minor variations in facial features. However, the average validation accuracy across all folds was 0.69, with an F1-Score of 0.65 and MSE of 0.37, indicating some variability in performance, possibly due to the less pronounced aging effects. As the age transformation increased to 15 years, the model's best accuracy in Fold 4 was 0.80, with 256 convolutional filters and 288 neurons in the first dense layer and a reduced number of 32 neurons in the second dense layer. The average validation accuracy was slightly lower at 0.68, with an F1-Score of 0.65 and MSE of 0.39 which might suggest that the model might have some difficulty generalizing to more complex or diverse cases. These values show that while the model does a decent job, it still struggles with predicting accurately when the data is more challenging. The higher MSE indicates that there are some errors in the predictions, and the model could improve with better tuning or adjustments to handle these cases more effectively. The 30-year transformation scenario presented the most considerable challenge due to the extensive changes in facial features over time, yet the model achieved its highest accuracy of 0.82 in Fold 4, with a different configuration of 64 convolutional filters and 288 and 160 neurons in the dense layers. The average validation accuracy for this scenario was 0.80, with an F1-Score of 0.71 and MSE of 0.35. The most effective kinship verification was observed in the 5-year scenario, where younger aging transformations made it easier for the model to

determine familial relationships.

Table 7.1: Comparison of Father-Children Relationship Accuracy Across Different Kinship Verification Models

Reference	Model/Approach	Accuracy
[21]	Cubic Norm and Kernel-based Bi-directional PCA	0.78
[6]	Kinship Verification via Reference List Comparison	0.82
[19]	Weighted Graph Embedding-Based Metric Learning	0.79
[4]	Cross-Generation Feature Interaction Learning	0.83
[20]	Age-uniform Feature Learning	0.76
[22]	Discriminative Subspaces of Color Components	0.77
[23]	Neighborhood Min Distance Descriptor	0.74
Proposed Model	Kinship Verification with Age Transformation	0.85

Table 7.3 shows a comparison with the state-of-the-art models where it is used age transformation as based to find kinship verification, while table 7.4 shows a comparison of the proposed model with other kinship verification models that also utilize age transformation. It can be pointed out that the proposed model, which incorporates age transformation techniques within a convolutional neural network (ResNet50) applied into a Siamese network, is specifically designed to enhance the kinship verification process by transforming facial images across different ages. This method allows the model to capture subtle changes in facial features due to aging, resulting in a high accuracy of 0.85 for the father-children relationship. This performance surpasses that of the "Kinship Identification using Age Transformation and Siamese Network" model, which achieved an accuracy of 0.78. Although the Siamese Network approach is effective in learning pairwise similarities, it does not explicitly focus on capturing age-specific transformations in a robust manner and also this research is using all the Families in the Wild dataset while the proposed model was trained with a subset of it. Similarly, the "Cross-Generation Feature

Interaction Learning" model, which focuses on learning interactions between features across different generations, achieved an accuracy of 0.83. While this approach does consider age-related changes, it does not apply explicit age transformations to the images, which may limit its ability to handle the nuanced variations in facial features over time as the proposed model does. Additionally, the "Age-uniform Feature Learning" model, which focus on learning features consistent across different ages to neutralize the age effect, achieved an accuracy of 0.76. This approach aligns with the concept of age transformation but focuses on maintaining feature uniformity rather than directly modeling transformations.

Table 7.2: Comparison of Father-Children Relationship Accuracy for Models Using Age Transformation

Reference	Model/Approach	Accuracy
[1]	Kinship Identification using Age Transformation and Siamese Network	0.78
[4]	Cross-Generation Feature Interaction Learning	0.83
[20]	Age-uniform Feature Learning	0.76
Proposed Model	Kinship Verification with Age Transformation	0.85

7.0.2. Mother-Children kinship verification analysis

The analysis of the kinship verification model for the mother-daughter relationship across various age transformation scenarios (5, 15, and 30 years) provides valuable insights into the model's ability to handle facial aging while maintaining accurate kinship identification. In the **15-year transformation scenario**, the model achieved its highest accuracy of **0.89** in Fold 4, utilizing 160 convolutional filters and 416 neurons in the first dense layer (unitsD1) and 160 neurons in the second dense layer (unitsD2). This configuration allowed the model

to capture the subtle facial features typically associated with a 5-year difference, demonstrating its capacity to effectively identify kinship when the facial aging effect is minimal. As medical research has suggested, facial features between parents and children tend to be more recognizable when the age difference is relatively small. Christenfeld & Hill [40] reported that people are better able to identify kinship between parents and younger children, which is consistent with the model’s higher accuracy in the 5-year transformation scenario.

For the **5-year transformation**, the model achieved its highest accuracy of **0.88**, with 192 convolutional filters and 416 neurons in the first dense layer. This adjustment in the network architecture reflects the need to capture more significant changes in facial features over time. While the model remains effective at identifying kinship, the challenge of facial feature variation due to more pronounced aging becomes evident in this scenario, as indicated by a slight decrease in performance metrics. The **average validation accuracy was 0.77**, with an F1-Score of **0.65**, and MSE of **0.35**. These metrics show that the model can still generalize well across different data subsets but faces challenges when dealing with more noticeable facial aging effects.

In the **30-year transformation scenario**, the model achieved its highest accuracy of **0.88** in Fold 1, with 96 convolutional filters and 416 neurons in the dense layers. This indicates that, despite the significant facial changes over a 30-year period, the model is still capable of capturing familial resemblance, though the overall performance showed signs of strain due to the complexity of handling such extensive age transformations. The **average validation accuracy** across all

folds was **0.78**, with an F1-Score of **0.66** and an MSE of **0.33**, showing that the model performs adequately when tasked with identifying kinship in the presence of more substantial facial variations.

Overall, the **mother-daughter relationship** scenario reinforces that the model performs best when age transformations are minimal, allowing it to capture familial resemblance more effectively. While the model remains robust even with larger transformations, further optimizations would be beneficial to handle more pronounced aging effects in the facial features, ensuring consistent accuracy across all age ranges.

Table 7.3: Comparison of Mother-Children Relationship Accuracy Across Different Kinship Verification Models

Reference	Model/Approach	Accuracy
[21]	Cubic Norm and Kernel-based Bi-directional PCA	0.62
[5]	Cross-Generation Kinship Verification with Sparse Discriminative Metric	0.80
[6]	Kinship Verification via Reference List Comparison	0.69
[19]	Weighted Graph Embedding-Based Metric Learning	0.82
[4]	Cross-Generation Feature Interaction Learning	0.82
[20]	Age-uniform Feature Learning	0.87
[22]	Discriminative Subspaces of Color Components	0.84
[23]	Neighborhood Min Distance Descriptor	0.83
Proposed Model	Kinship Verification with Age Transformation	0.89

Table 7.3 shows a comparison between the proposed model and other state-of-the-art kinship verification models. The table shows that the proposed model for kinship verification, which incorporates age transformation techniques, demonstrated a strong performance with an accuracy of 0.89 for the mother-children relationship. The proposed model shows the highest-performing approach among the rest of the state-of-the-art models. This highlights the effectiveness of the proposed model in

handling the task of identifying kinship relationships when facial features change due to age progression.

When compared to earlier approaches like the Cubic Norm and Kernel-based Bi-directional PCA, which achieved a much lower accuracy of 0.62, the proposed model shows a substantial improvement. This increase in performance can be attributed to the integration of age transformation, which enhances the model’s ability to capture changes in facial features over time. Similarly, models like Cross-Generation Kinship Verification with Sparse Discriminative Metric (0.80) and Kinship Verification via Reference List Comparison (0.69) fall short of the accuracy achieved by the proposed model, further demonstrating it got a better performance in dealing with age-based facial feature transformations.

Other high-performing models, such as Age-uniform Feature Learning (0.87) and Discriminative Subspaces of Color Components (0.84), also achieved competitive results, but the proposed model’s use of explicit age transformations offers an advantage. These transformations allow the model to better account for the natural aging process, which is critical in kinship verification tasks involving individuals of significantly different ages.

It is important to mention that many of the models referenced in this comparison were tested on different datasets, including the Cornell, KinFaceW-I, and KinFaceW-II datasets. These datasets, while widely used in kinship verification research, present different challenges compared to the FIW dataset, which was used to validate the proposed model. The FIW dataset includes a more diverse set of familial relationships, making it a more challenging for kinship

verification. Despite this, the proposed model’s ability to maintain high accuracy demonstrates its robustness and adaptability to real-world kinship verification tasks, where age progression plays a significant role.

Table 7.4: Comparison of Mother-Children Relationship Accuracy for Models Using Age Transformation

Reference	Model/Approach	Accuracy
[1]	Kinship Identification using Age Transformation and Siamese Network	0.74
[4]	Cross-Generation Feature Interaction Learning	0.82
[20]	Age-uniform Feature Learning	0.87
Proposed Model	Kinship Verification with Age Transformation	0.89

Table 7.4 compares the proposed model with other state-of-the-art models that also includes image transformation as part of their solutions. The table illustrates that the proposed model achieves an accuracy of 0.89 in the mother-children relationship, better than the accuracies of models like Kinship Identification using Age Transformation and Siamese Network (0.74) and Cross-Generation Feature Interaction Learning (0.82). Both models have shown strong performances, particularly in extracting features from transformed images. However, the proposed model benefits from an enhanced architecture that combines age transformation with a Siamese network tuned for discriminative feature extraction, allowing for better recognition of familial relations, even across varying age gaps.

In comparison, the Age-uniform Feature Learning approach also achieves high accuracy (0.87), emphasizing the importance of minimizing the age discrepancy between parent and child images. This model implements a GAN-based transformations to unify the age of the images, similar to the Life Span Age Transformation (LATS) algorithm used in the proposed model, but the slight

improvement in accuracy of the proposed model suggests a more effective handling of age-related facial feature changes.

It’s also important to mention that the models Age-uniform Feature Learning and Cross-Generation Kinship Verification, were tested on datasets like KinFaceW-I, KinFaceW-II, and Cornell KinFace, which differ from the FIW dataset used in the proposed model. This makes direct comparisons somewhat challenging, as the data characteristics (e.g., resolution, pose variation, image quality) vary across datasets. Despite this, the proposed model shows robust performance across various mother-children relationships and demonstrates its capacity to generalize well to the challenging FIW dataset.

7.0.3. General results analysis of the proposed model

In comparing the best accuracies achieved for the father-children and mother-children relationships, it is observed different trends in performance depending on the age transformation applied. For the father-children relationship, the highest accuracy of 0.85 occurred when the images were transformed to represent a 5-year age difference, as seen in Figure 7.1. In contrast, the mother-children relationship achieved its best accuracy of 0.89 when the images were transformed to reflect a 15-year age difference, as seen in Figure 7.2.

One of the key factors influencing the accuracy of kinship verification models is the age of the individuals being compared. Several studies have shown that facial features of younger individuals tend to exhibit more childish characteristics, such as rounder faces, larger eyes, and softer contours. These traits are typically less

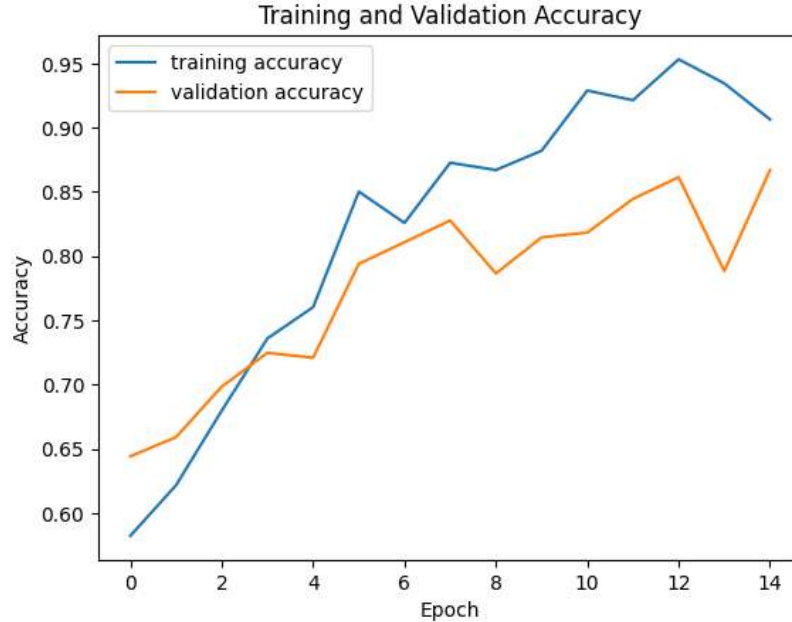


Figure 7.1: Best accuracy got on 5-years old transformation for relation Father-Children

distinct and more similar across family members during early ages, making it easier for models to detect familial resemblance. One study on kin recognition emphasizes that kin are more likely to be recognized through phenotypic traits like facial features, particularly when individuals are younger and share more similarities with their family members [41]. Also, humans naturally have an enhanced ability to detect relatedness based on facial features in younger individuals. This ability is due to the fact that their features haven't yet fully developed into distinct adult traits, making familial resemblance more apparent [42]. Studies also has shown how facial resemblance plays a role in kinship recognition, indicating that younger individuals exhibit more pronounced and distinguishable facial features that make it easier to detect familial relationships [43]. Also, there is a study that explored children's

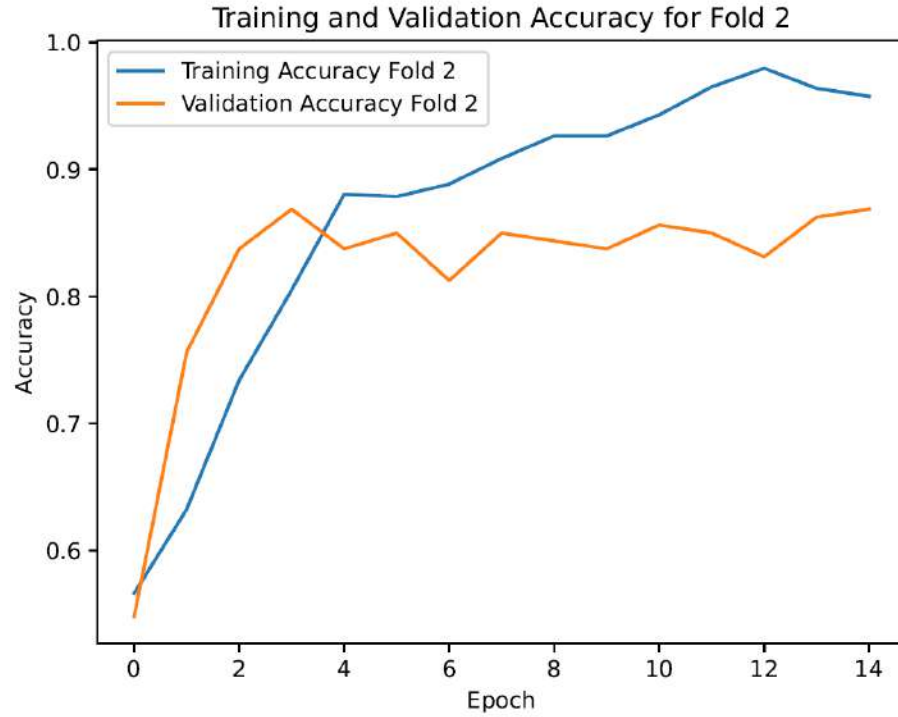


Figure 7.2: Best accuracy got on 15-years old transformation for relation Mother-Children.

ability to detect kinship through facial resemblance, reinforcing the idea that facial features of younger individuals are more recognizable and similar within families [44].

The image 7.3 shows the accuracy values got for each age transformation on the father-children and mother-children relations. The model achieved the highest accuracy of 0.85 in the 5-year transformation scenario for father-children relationships. This suggests that the model is most effective when the age difference is minimal, allowing it to detect familial resemblance more accurately. As the age transformation increased to 15 years, the accuracy dropped to 0.80. This decline indicates that the model finds it more challenging to identify kinship

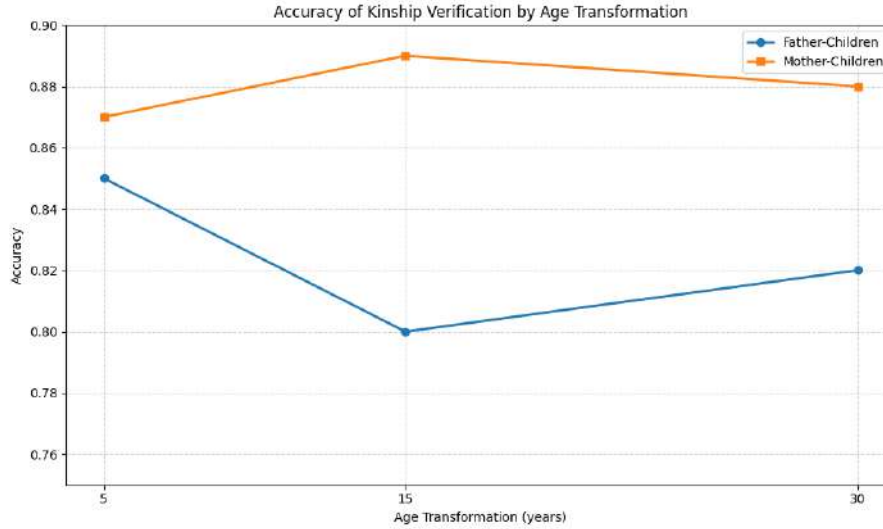


Figure 7.3: Comparison of Father-Mother accuracy through age transformations.

features as the age gap increases, likely due to more pronounced changes in facial features over time. Interestingly, the accuracy improved slightly to 0.82 in the 30-year transformation scenario, suggesting that the model could still capture some key familial features despite significant aging effects.

In contrast to the father-children results, image 7.3 shows that the mother-children relationship showed the highest accuracy in the 15-year transformation scenario, achieving 0.89. This result suggests that the model is better at capturing familial features in this scenario, possibly due to more pronounced maternal facial traits becoming evident as children age, this result could be related with studies that support the idea that humans possess the ability to detect kinship through facial traits, emphasizing that these abilities may be affected by the degree of relatedness and could be more pronounced when it comes to maternal features as individuals age [45]. This finding aligns with the concept

that the model is better at capturing familial features, especially maternal traits, as children grow older, as seen in the 5 years transformation shown on figure 7.4.

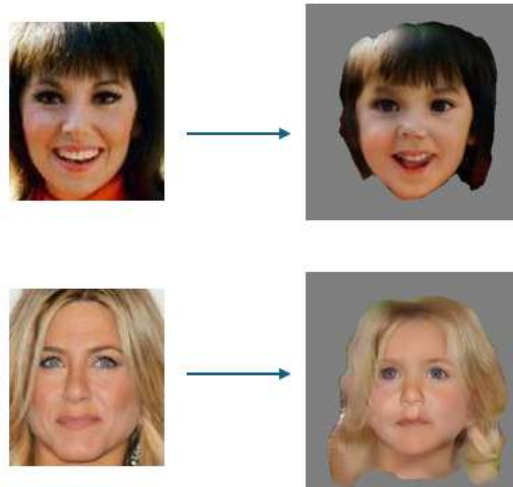


Figure 7.4: Transformation mother-kid to 5 years old.

The accuracy in the 5-year transformation scenario was slightly lower at 0.87. Despite this, it still demonstrates the model’s ability to detect kinship relationships effectively when the age difference is minimal. The accuracy remained high at 0.88 in the 30-year transformation scenario, indicating that the model maintained strong performance despite the larger age gap.

The notion that facial features are more similar in early ages is also supported by studies which suggests that younger individuals retain features that are less altered by external factors such as lifestyle or environmental exposure. In that way, when comparing kinship relations at younger ages, models are more likely to detect similarities in these neutral, “childish” facial features [40]. Kin recognition signals are more evident in younger faces, which supports the notion that kinship

verification is more effective when using images of individuals at an earlier age, as facial features are less altered by external factors [46].

Given this, it is logical to suggest that kinship verification models perform more effectively when images from earlier ages are used like 5 years old or 15 years old as proved using the proposed model, as they are less impacted by variations introduced by aging. The results from the proposed model validate the fact that young features in faces makes easier to check for kin relations, as it was achieved higher accuracy in detecting kinship when younger age transformations were applied to the images.

7.0.4. Contributions

Some of the highlights of the analysis and contributions of this research are listed:

- The proposed model demonstrates improvements in accuracy while handling age-related variations, where the highest accuracy of 0.85 occurred when the images were transformed to represent a 5-year age difference in father-children relation while, the mother-children relationship achieved its best accuracy of 0.89 when the images were transformed to reflect a 15-year age difference.
- The model effectively captures changes in facial features with a 5-year and 15-year age difference, making it suitable for identifying kinship with younger age differences.
- The model effectively captures changes in facial features when there is a 5-year and 15-year age difference, making it particularly suitable for identifying kinship relations with younger age groups. This result aligns with theories such

as those of Christenfeld & Hill [40], who suggested that youthful facial features are easier to match with parents. Their research demonstrated that people could accurately pair fathers with their 1-year-old children, underscoring the advantage of using youthful features for kinship verification.

- The approach of applying direct age transformations proves to be more effective in kinship verification, as it provides a detailed and dynamic understanding of how aging impacts facial features, allowing for more accurate identification of familial relationships across different age spans.
- The model performs optimally when images are transformed to represent a 5-year age difference in father-children relations and a 15-year age difference in mother-children relations. These findings suggest that youthful and adolescent facial features are more reliable for kinship verification than older ones. The ability of the model to effectively distinguish familial traits in early and adolescent stages strengthens its reliability across varying age groups.
- The proposed approach of applying direct age transformations proved to be more effective than other state-of-the-art models for kinship verification. By focusing on dynamic facial feature changes caused by aging, the model allows for more precise identification of familial relationships across different age ranges. This is especially evident when compared with other models, as seen in Table 7.5, where the proposed model achieved a global accuracy of 0.87.
- Additionally, the model benefits from using a comprehensive dataset (FIW), which contains diverse family relations and is tested across various familial

scenarios. This enhances the model’s generalization capabilities and practical application, making it a valuable contribution to the field of kinship verification.

Table 7.5: Comparison of Global Accuracy Across Different Kinship Verification Models

Reference	Model/Approach	Accuracy
[21]	Cubic Norm and Kernel-based Bi-directional PCA	0.78
[6]	Kinship Verification via Reference List Comparison	0.82
[19]	Weighted Graph Embedding-Based Metric Learning	0.79
[4]	Cross-Generation Feature Interaction Learning	0.83
[20]	Age-uniform Feature Learning	0.76
[22]	Discriminative Subspaces of Color Components	0.77
[23]	Neighborhood Min Distance Descriptor	0.74
[24]	Deep Relational Features	0.81
Proposed Model	Kinship Verification with Age Transformation (Proposed Model)	0.87

The contributions of this work underscore its significance in advancing the field of kinship verification, particularly in its ability to handle age-related variations with improved accuracy. The proposed model demonstrated improves in accuracy, achieving a 0.85 in father-children relationships with a 5-year age difference and 0.89 in mother-children relationships with a 15-year difference. This reflects the model’s strength in identifying kinship in younger age groups, aligning with theories that suggest youthful facial features are more indicative of familial ties. The model present a global accuracy of 0.87 which in comparison to similar models that implements age transformation improved this aspect. Additionally, the use of the FIW dataset highlights the model’s versatility and its potential to generalize across diverse family relationships, making it a valuable and innovative contribution to the field of kinship verification.

7.0.5. Limitations

The limitations encountered during the development of the proposed kinship verification model reflect some challenges inherent in working with real-world data. While these factors had an impact on the overall analysis, they also provided insights into areas for future refinement. The FIW dataset, for example, had some instances of incomplete or lower-quality images, which influenced the transformation process. Additionally, the complexity of diverse family relationships in the dataset introduced certain challenges when focusing specifically on mother-children and father-children relationships. Despite these aspects, the model’s ability to adapt and perform effectively underscores its potential and offers valuable learning experiences for future improvements.

The discussion presented highlights the strengths, contributions, and challenges of the proposed kinship verification model across different age transformation scenarios. The model demonstrated strong performance in identifying familial relationships, particularly in younger age transformations, reinforcing the effectiveness of incorporating age progression techniques and the fact that kin features are more likely to be similar between family members when photos are at young ages. One of the key contributions of this research is the model’s ability to handle age-related variations, capturing changes in facial features with 5-year and 15-year age differences, which proved to be effective for kinship verification in younger age groups. This aligns with studies suggesting that youthful features are more indicative of familial relationships, as demonstrated by the high accuracy achieved in father-children and mother-children scenarios. Despite certain

limitations, such as data quality issues and the manual intervention on the database to select the best images, the model’s enhanced generalization capabilities and use of the comprehensive FIW dataset make it a valuable contribution to the field of kinship verification research.

Conclusions and Future Work

This chapter presents the final conclusions obtained through the research effort and the future work that can be derived. Through the obtained results are highlighted, the model’s strengths and limitations in identifying father-children and mother-children pairs under different conditions, contributing to a deeper understanding of its robustness and adaptability in real-world applications.

8.1. Conclusions

This work introduced a kinship verification model that integrates age transformation techniques with deep learning, specifically applied to the Families in the Wild (FIW) dataset. The results demonstrate that incorporating age transformations significantly enhances the model’s ability to capture familial features, achieving notable success in identifying kinship relationships across the age ranges of 5, 15 and 30 years. The results show that the model exceeds in father-children relationships with a 5-year transformation and mother-children relationships with a 15-year transformation. This reveals that age transformation techniques are important in highlighting distinct familial features that changes during the years.

One of the key achievements of this study was confirming that age transformation plays a vital role in kinship verification, offering an approach to understanding how familial features evolve with time. The model’s success in capturing these features reaffirms studies suggesting that kinship similarities are more apparent at certain ages, especially when comparing younger individuals. This finding is supported by

the higher accuracy rates observed in the 5-year and 15-year transformations.

The proposed approach was proven to be more effective than other state-of-the-art methods, achieving a competitive accuracy of 0.87. This suggests that incorporating age-related changes into the learning process offers a deeper insight into familial resemblance, which leads to more accurate verification.

Additionally, the model’s robustness across different age transformation scenarios highlights its potential applicability to real-world scenarios where kinship verification is needed. By leveraging the comprehensive FIW dataset, which contains diverse family images, the model shows strong accuracy, making it a valuable contribution to the field of kinship verification research.

For each of the specific objectives proposed in this research, the next list summarizes the key findings and insights obtained per objective, demonstrating how the proposed model effectively addressed the goals of this study.

- **Objective 1:** To determine the limitations within the kinship verification research area with the aim of finding areas for improvement.
 - The research identified that one of the primary limitations in the kinship verification field is the challenge of accurately capturing familial features across different age ranges, as many existing models struggle to account for the natural aging process. This gap shows the need for integrating age transformation techniques to improve model performance, as evidenced by the results of the proposed approach in handling age-related variations.
 - Another significant limitation found was the quality and diversity of available datasets, which often contain images with poor resolution,

inconsistent poses, or varying lighting conditions. Addressing this limitation through the proposed model’s use of age transformation and model tuning techniques demonstrated that it is possible to improve kinship verification accuracy despite dataset constraints.

- **Objective 2:** To design a kinship verification algorithm integrating age transformation techniques and model tuning that enhances the algorithm’s capability to identify familial relationships.
 - The proposed algorithm successfully integrated age transformation techniques, which enhanced its ability to identify familial relationships more accurately, especially in the 5-year and 15-year transformations for father-children and mother-children. This integration allowed the model to capture changes in facial features that traditional models may overlook, proving the effectiveness of incorporating age transformation in kinship verification tasks.
 - Model tuning played a crucial role in enhancing the algorithm’s performance, enabling it to adapt to different familial features across different age ranges. This tuning process demonstrated that carefully adjusting model parameters can significantly improve kinship verification outcomes.
- **Objective 3:** To evaluate the results of the proposed algorithm by implementing metrics applied to scenarios that focus on age transformation and model tuning using the Families in the Wild (FIW) dataset.

- The evaluation using the FIW dataset showed that the proposed algorithm achieved competitive accuracy compared to state-of-the-art models, with a global accuracy of 0.87. This result confirms that the integration of age transformation and model tuning is effective in improving kinship verification performance.
- The metrics applied in different age transformation scenarios revealed that the model is most effective in younger age ranges, particularly in the 5-year and 15-year transformations for father-children and mother-children relation, where it captured familial relationships with high accuracy. This finding validates the hypothesis that age transformation techniques, combined with model tuning, can significantly enhance the accuracy of kinship verification models.

This work presents results that fit with the proposed objectives, also it effectively demonstrated the importance of integrating age transformation techniques for enhancing kinship verification, reinforcing the idea that early-age facial features, especially in the 5- and 15-year transformations, are more easily recognized by models as kinship indicators showing a clear improvement over state-of-the-art methods.

8.2. Future Work

The proposed kinship verification model has shown significant progress, but there are areas where further exploration and refinement could lead to even more

robust and accurate results. Given the limitations encountered during this study, including the use of a single dataset and certain architectural constraints, future work can address these areas to enhance the model’s performance and scope. Expanding the research to cover more datasets, relationships, and advanced methodologies would provide a different perspective on how kinship can be verified across different populations and conditions. Some of the ideas to expand in future research are listed:

- **Larger datasets:** Utilize larger datasets with better image quality could significantly enhance the model’s robustness and performance. Neural networks tend to perform better with more data, as it reduces the risk of overfitting and allows the model to learn from a wider range of examples [5]. Creating a new family relation dataset with high-quality images would provide more training data, leading to improved accuracy and generalization. Data augmentation could also be employed to artificially increase dataset size, which has been shown to enhance model performance in various deep learning tasks [6].
- **Include different kinship relations:** Including other kinship relationships like siblings or grandparents would offer new insights into how familial features manifest across different family relations. Include different family relations helps in developing models that are more capable of handling complex relationships, as different family members share varying degrees of similarity [19].

- **Addressing dataset quality:** Implementing advanced data augmentation techniques can improve image quality and help the model handle poor-quality images more effectively. Data augmentation strategies, such as adding noise, adjusting brightness, or using GANs (Generative Adversarial Networks) to generate higher-quality images, have been proven to enhance model performance [22]. These techniques could reduce the impact of low-quality images and improve the overall accuracy of kinship verification.
- **Incorporating additional features:** Combining facial features with other biometric data, such as voice or gait, can provide a more comprehensive understanding of kinship verification. Research has shown that multimodal biometric systems are often more accurate than single-modality systems, as they provide complementary information. Integrating multiple features would likely increase the model's robustness and improve its kinship verification capabilities [23].
- **Work with different age ranges below 5 years:** It was proved that early facial features are better to find kin relations in between family members and also, it was tested in this research the scenarios of 5, 15 and 30 years. This work aims to keep research on early ages above years, it was proved that 5 years transformation was better than older ages so, test on ages around 1 to 5 years will be an option of accuracy improvement.
- **Improve the hardware where the model was tested:** This work was tested on a Colab instance. Figure 9.1 shows the configuration used during the

evaluation process. The Colab instance runs with Python3, a T4 GPU and High-RAM activated. Initial work on the research used a free Colab account but, it does not support the training process due to hardware lack. Some improvements with better processing will be increasing the amount of epochs and amount of convolutional layers on the ResNet50 model. Same with the amount of trials when working with the model tuning, it can be increased to more than 3 trials.

How to address dataset challenges, include multiple family relationships and mix different features not just facial ones applied on a model that uses age transformation and hyperparameter tuning is a proposing direction for future research. As a final note, it is suggested that the Master's Program in Computer Science consider opportunities to strengthen research infrastructure, which is essential for the development of projects like this one. Lack of infrastructure such as servers and data storage are crucial to improve the results of research. Likewise, reinforcing the commitment and support of the program's advisors is important, as their guidance and encouragement are decisive elements that help students progress and successfully complete their research and finally graduate from this program.

9.1. Source Code

Database and code can be found in this GitHub repository by clone it using the web URL <https://github.com/ppiedra/FIWDDatabase.git>. The URL to the repository and branch <https://github.com/ppiedra/FIWDDatabase/tree/final>. This repository is divided into four branches: main, backup, testing and final. Use

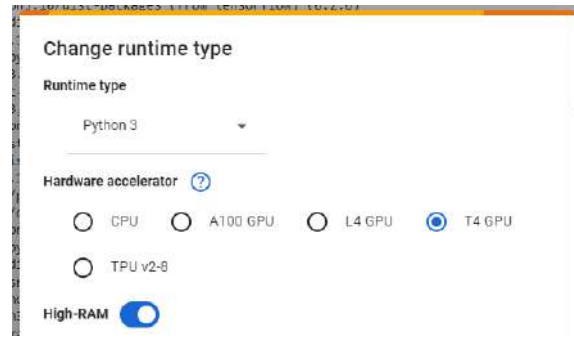


Figure 9.1: Colab Pro configuration.

final branch to copy the latest version of the code and dataset. The branch comes with the next data:

- FinalReports: this directory is used by the model to store the final reports at the end of each fold.
- train-faces: This is the original Families in the Wild dataset.
- train-faces30Clean: This is the 144 families cleaned manually. Here the original photos with poor quality were removed.
- train-facesAge<5,15,30>B1: are the databases transformed into the ages 5, 15 and 30 using the LATS transformation algorithm.

9.1. Source Code

- `cleancodekinshipidentificationusingsiamesenetwork.py`: is the model's code, use Colab Pro to run it.
- `train-pairs.csv`: the original csv used by Families in the Wild dataset to match photos with families. It also dictates the relations each photo has in the family.

Figure 9.1 shows the configuration used during the evaluation process. The Colab instance runs with Python3, a T4 GPU and High-RAM activated. Initial work on the research used a free Colab account but, it does not support the training process due to hardware lack.

References

- [1] Arshad Abbas and Muhammad Shoaib. Kinship identification using age transformation and siamese network. *PeerJ Computer Science*, 2022.
- [2] Nour El Houda Bouakal, Messaoud Hettiri, Abdelhakim Chergui, Abdelkrim Ouafi, Azeddine Benlamoudi, and Salah Eddine Bekhouche. Kinship verification system based on the color spaces analysis. *2022 7th International Conference on Image and Signal Processing and their Applications, ISPA 2022 - Proceedings*, 2022.
- [3] Xiaoke Zhu, Changlong Li, Xiaopan Chen, Xinyu Zhang, and Xiao-Yuan Jing. Distance and direction based deep discriminant metric learning for kinship verification. *ACM Transactions on Multimedia Computing, Communications and Applications*, 2023.
- [4] Guan-Nan Dong, Chi-Man Pun, and Zheng Zhang. Kinship verification based on cross-generation feature interaction learning. *IEEE Transactions on Image Processing*, 2021.
- [5] Shuyang Wang, Zhengming Ding, and Yun Fu. Cross-generation kinship verification with sparse discriminative metric. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, page 2783 – 2790, 2019.
- [6] Wenna Zheng and Junlin Hu. Kinship verification via reference list comparison. *Lecture Notes in Computer Science (including subseries Lecture Notes in*

- Artificial Intelligence and Lecture Notes in Bioinformatics*), page 384 – 391, 2021.
- [7] Wei Wang, Shaodi You, Sezer Karaoglu, and Theo Gevers. Kinship identification through joint learning using kinship verification ensemble. *ArXiv*, 2020.
 - [8] Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Säckinger, and Roopak Shah. Signature verification using a "siamese" time delay neural network. In *Advances in Neural Information Processing Systems*, 1994.
 - [9] Jiwen Lu, Junlin Hu, and Yap-Peng Tan. Discriminative deep metric learning for face and kinship verification. *IEEE Transactions on Image Processing*, page 4269 – 4282, 2017.
 - [10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
 - [11] F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.
 - [12] Tom O’Malley, Elie Bursztein, James Long, François Chollet, Haifeng Jin, Luca Invernizzi, et al. Kerastuner, 2019.
 - [13] Ruogu Fang, Kevin D. Tang, Noah Snavely, and Tsuhan Chen. Towards computational models of kinship verification. 2010.
 - [14] Ming Shao, Siyu Xia, and Yun Fu. Genealogical face recognition based on ub kinface database. 2011.
 - [15] N. Zhang, J. Wang, S. Shan, and X. Chen. Kinship verification with deep convolutional neural networks. 2015.
 - [16] Ruogu Fang, Kevin D. Tang, Noah Snavely, and Tsuhan Chen. Kinship classification by modeling facial feature heredity. 2013.
 - [17] Joseph P. Robinson, Ming Shao, Yue Wu, and Yun Fu. Families in the wild (fiw): Large-scale kinship image database and benchmarks. In *Proceedings of the 2016 ACM on Multimedia Conference*, pages 242–246. ACM, 2016.
 - [18] Naman Kohli. Automatic kinship verification in unconstrained faces using deep learning. 2019.

- [19] Jianqing Liang, Qinghua Hu, Chuangyin Dang, and Wangmeng Zuo. Weighted graph embedding-based metric learning for kinship verification. *IEEE Transactions on Image Processing*, page 1149 – 1162, 2019.
- [20] Liang Zheng, Ya Chen, Xiaopan Chen, and Fengbin Zheng. Age-uniform feature learning for image-based kinship verification. *ACM International Conference Proceeding Series*, page 65 – 71, 2021.
- [21] Mohammad Mahdi Dehshibi and Jamshid Shanbehzadeh. Cubic norm and kernel-based bi-directional pca: toward age-aware facial kinship verification. *Visual Computer*, page 23 – 40, 2019.
- [22] Oualid Laiadi, Abdelmalik Ouamane, Elhocine Boutellaa, Abdelhamid Benakcha, Abdelmalik Taleb-Ahmed, and Abdenour Hadid. Kinship verification from face images in discriminative subspaces of color components. *Multimedia Tools and Applications*, page 16465 – 16487, 2019.
- [23] Abdellah Sellam and Hamid Azzoune. Neighborhood min distance descriptor for kinship verification. *Multimedia Tools and Applications*, page 20861 – 20880, 2020.
- [24] Jingyun Liang, Jinlin Guo, Songyang Lao, and Jue Li. Using deep relational features to verify kinship. *Communications in Computer and Information Science*, page 563 – 573, 2017.
- [25] Raunak Sinha, Mayank Vatsa, and Richa Singh. Familygan: Generating kin face images using generative adversarial networks. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, page 297 – 311, 2020.
- [26] F. Dornaika, I. Arganda-Carreras, and O. Serradilla. Transfer learning and feature fusion for kinship verification. *Neural Computing and Applications*, page 7139 – 7151, 2020.
- [27] Madhu Oruganti, Toshantal Meenpal, Saikat Majumdar, and Sri Ramesh Naidu Kusu. Deep learning model for facial kinship verification using childhood images. *2023 International Conference for Advancement in Technology, ICONAT 2023*, 2023.
- [28] Min Xu and Yuanyuan Shang. Kinship measurement on face images by structured similarity fusion. *IEEE Access*, page 10280 – 10287, 2016.

- [29] Ximiao Zhang, Min Xu, Xiuzhuang Zhou, and Guodong Guo. Supervised contrastive learning for facial kinship recognition. *Proceedings - 2021 16th IEEE International Conference on Automatic Face and Gesture Recognition, FG 2021*, 2021.
- [30] Xiaoqian Qin, Xiaoyang Tan, and Songcan Chen. Tri-subjects kinship verification: Understanding the core of a family. *Proceedings of the 14th IAPR International Conference on Machine Vision Applications, MVA 2015*, page 580 – 583, 2015.
- [31] Eran Dahan, Yosi Keller, and Shahar Mahpod. Kin-verification model on fiw dataset using multi-set learning and local features. *RFIW 2017 - Proceedings of the 2017 Workshop on Recognizing Families In the Wild, co-located with MM 2017*, page 31 – 35, 2017.
- [32] Aarti Goyal and Toshnall Meenpal. Kinship verification using multi-level dictionary pair learning for multiple resolution images. *Pattern Recognition*, 2023.
- [33] Yan-Guo Zhao, Zhan Song, Feng Zheng, and Ling Shao. Learning a multiple kernel similarity metric for kinship verification. *Information Sciences*, page 247 – 260, 2018.
- [34] Haibin Yan, Jiwen Lu, and Xiuzhuang Zhou. Prototype-based discriminative feature learning for kinship verification. *IEEE Transactions on Cybernetics*, page 2535 – 2545, 2015.
- [35] Yong Li, Jiabei Zeng, Jie Zhang, Anbo Dai, Meina Kan, Shiguang Shan, and Xilin Chen. Kinnet: Fine-to-coarse deep metric learning for kinship verification. *RFIW 2017 - Proceedings of the 2017 Workshop on Recognizing Families In the Wild, co-located with MM 2017*, page 13 – 20, 2017.
- [36] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf. Deepface: Closing the gap to human-level performance in face verification. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.
- [37] Claes Wohlin, Per Runeson, Martin Höst, Magnus C. Ohlsson, Björn Regnell, and Anders Wesslén. *Experimentation in Software Engineering*. Springer, 2012.
- [38] Roy Or-El, Soumyadip Sengupta, Ohad Fried, Eli Shechtman, and Ira Kemelmacher-Shlizerman. Lifespan age transformation synthesis. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020.

- [39] Jiang Suo, Xi Zhu, Shiguang Shan, Xilin Chen, and Wen Gao. A concatenational graph evolution aging model for face age progression. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2012.
- [40] Nicholas J. Christenfeld and Emily A. Hill. Whose baby are you? *Nature*, 1995.
- [41] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Deep sparse rectifier neural networks. 2011.
- [42] Jill M. Mateo. Kin recognition in ground squirrels and other animals. *Ethology*, 108(9):765–787, 2002.
- [43] M. F. Dal Martello and L. T. Maloney. Where are kin recognition signals in the human face? *Journal of Vision*, 2006.
- [44] J. Kaminski, E. Gentaz, and K. Mazens. Developmental changes in children’s sensitivity to facial resemblance: Is kinship detection a vital component of children’s cognitive development? *British Journal of Developmental Psychology*, 2012.
- [45] G. Kaminski, S. Dridi, C. Graff, E. Gentaz, and E. Monfardini. Human ability to detect kinship in strangers faces effects of the degree of relatedness. *Proceedings of the Royal Society B: Biological Sciences*, 2009.
- [46] L. M. DeBruine, B. C. Jones, A. C. Little, and D. I. Perrett. Kin recognition signals in adult faces. *Vision Research*, 2009.
- [47] UNICEF USA. More than half of ukraine’s children displaced by war. *UNICEF USA*, 2022.
- [48] Tom O’Malley, Elie Bursztein, James Long, François Chollet, Haifeng Jin, Luca Invernizzi, et al. Kerastuner, 2019.
- [49] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016.
- [50] James F. Leckman. Displaced children, youth, and families traumatized by war and violent conflict in ukraine and beyond. *Yale School of Medicine*, 2022.
- [51] Gwenaél Kaminski, Soumeiya Dridi, Camille Graff, and Edouard Gentaz. Human ability to detect kinship in strangers. *Perception*, 38(5):659–666, 2009.