Manu Varma

Kin Recognition Using Weighted Graph Embeddings

Computer Science Tripos – Part II

St John's College

March 22, 2021

Declaration

I, Manu Varma of St. John's College, being a candidate for Part II of the Computer Science Tripos, hereby declare that this dissertation and the work described in it are my own work, unaided except as may be specified below, and that the dissertation does not contain material that has already been used to any substantial extent for a comparable purpose.

Signed Manu Varma

Date March 22, 2021

Proforma

Name:	**Manu Varma**
College:	**St John's College**
Project Title:	**Kin Recognition Using Weighted Graph Embeddings**
Examination:	**Computer Science Tripos – Part II, June 2021**
Word Count:	**TBA**
Project Originator:	**The Dissertation Author**
Supervisor:	**Daniel Bates**

Original Aims of the Project

Work Completed

Special Difficulties

Contents

Chapter 1

Introduction

Kinship recognition is the way of recognizing whether two given people are related to each other or not based on how they look. This is an evolutionary trait in humans as it is advantageous to inclusive fitness for an organism to be able to recognize which of their neighbors were close relatives [2]. Thus, it stands to reason that the ability to recognize kinship relationships has evolved in humans. In humans, specifically, facial resemblance is expected to serve as an indicator of kinship as we have seen that strangers are able to match photographs of mothers to their infants without any prior contact with any of the family [6].

Computational kinship recognition is the field of computationally figuring out kinship relationships between people without any prior knowledge of the family.

1.1 Problem Overview

There are multiple problems in the field of computational kin recognition, in which case I will focus on one of them. The first of which takes as input a pair of images and a proposed kinship relationship, for example father-daughter, and recognizes whether the relationship exists. These relationships tend to be more commonly parent-child relationships as opposed to sibling-sibling relationships, for example, which are less commonly used. A further extension of the main kin recognition problem is Tri-Subject Kinship which takes 3 images, two parents and a child, and determines if they are related or not. These problems are the ones that are being looked at in this dissertation.

However, some other major problems in the field include search-and-retrieval and family classification. Search-and-retrieval is the problem which takes an image of a person as input and searches through a database to find people with whom they are most likely to be related to. This outputs a list of these people that they could be related to. Family classification deals with a similar task of taking an image of a person as input and figuring out which family they may belong to.

1.2 Motivation

Accurate kinship recognition software has multiple applications in humanitarian issues. With regards to humanitarian issues, by using kin recognition, we can better recognize missing children and match them with their parents [9]. It can also be used for stopping human traffickers from claiming they are family members of the victim or to reunite families across refugee camps.

Furthermore, there are potential use-cases in social media and also poses privacy protections.

1.3 Related Work

One of the first papers in the field of computational kinship recognition used color, facial parts, facial distances and a Histogram of Gradients vector as the features for each image. Using these features, K-Nearest-Neighbors and an SVM were used in order to classify the image pairs into true and false parent-child pairs [1]. A classification accuracy of 70.67% was obtained overall and an SVM with a radial basis kernel obtained an accuracy of 68.6%.

Other approaches include using deep learning to solve the problem. One paper used a Siamese CNN approach by putting both of the face images in the pair through a SqueezeNet network which was trained on VGGFace2 which creates a feature vector for each image [7]. Using a similarity criterion, a new feature vector is created using the two feature vectors and a fully connected layer and a sigmoid activation function creates the predicted similarity score. This approach yielded an average test accuracy of 67.66% over all of the relationships that were in the dataset. Another paper that used a Siamese approach aimed to solve both the standard kinship verification problem and the Tri-Subject problem [10]. Both networks for the problems had three stages, a feature extraction stage, which used the ResNet50 or SENet50 models pretrained on VGGFace2 to extract the features from each face into a vector, a feature fusion stage which combines the feature vectors together, and a similarity quantization stage to get the similarity score. They then use a jury system to fuse together models which allows them to achieve an accuracy of 75.9%.

However, a prominent approach that is taken includes using metric learning to solve the problem. For example, the paper on Neighborhood Repulsed Metric Learning [5] uses a supervised metric learning approach. Using the approach, they aim to find a metric which minimizes the distance between the vectors of images that have a kinship relationship and maximize the distances between the vectors of pairs of images that don't have a kinship relationship. Furthermore, the paper that will be implemented in this dissertation, using Weighted Graph Embedding-Based Metric Learning, involves a metric learning approach to the problem [3].

1.4 Project Overview

The project deals with a 2019 method using Weighted Graph Embedding-Based Metric Learning [3], or WGEML for short, and aims to implement the paper and verify the accuracies that were obtained. The following is shown in the dissertation:

1. The algorithm, WGEML, is implemented and accuracies that are within an acceptable range are obtained, as shown in section 4.2.

2. Ablation studies on the face descriptors are performed and the results are in section 4.4.1.

3. We use the models that were created to discover biases in the datasets that were used and find out the impact that has overall in section 4.3.

4. We further test the face descriptors by replacing VGG with a smaller CNN in which the implementation is discussed in section 3.7.

Chapter 2 explains much of the needed technical background for the implementation, from what a Neural Network is and what metric learning is to each of the face descriptors used and what face descriptors are. Chapter 3 will then discuss the specifics of how WGEML and the extension that used a different network than VGG was implemented. Chapter 4 then discusses all of the results that were obtained from experimentation with WGEML and the implications of the results.

Chapter 2

Preparation

2.1 Convolutional Neural Networks

We must first

2.2 Face Detection

2.3 Face Descriptors

We wish to be able to create a mapping from a colored image into a vector to make computations easier and to be able to determine similarity between images, which we do using the distance. These are called image descriptors for general images. When we try and map face images to vectors, we call these face descriptors. We want these face descriptor mappings to be able to match the same person in different poses and illuminant geometries to face descriptors that are close to each other in distance. Thus, these mappings try and capture color, texture, and shapes, for example. There are multiple face descriptors that can be made of a face image, each of which extracts different features from the face. We use the Local Binary Patterns, Histogram of Gradients, Scale-Invariant Feature Transform and VGG face descriptors to extract features from each face.

2.3.1 Local Binary Patterns

The Local Binary Patterns (LBP) face descriptor takes an image

2.3.2 Histogram of Gradients

2.3.3 Scale-Invariant Feature Transform

2.3.4 VGG

2.3.5 CifarNet

2.4 K-Nearest Neighbors

2.5 Metric Learning

2.6 Principle Component Analysis

2.7 Requirements Analysis

2.8 Software Engineering Practices

2.8.1 Starting Point

Before I started my project, I had knowledge in Python and surface-level knowledge of Keras and Tensorflow. I had also worked with face recognition before. Furthermore, I had used Numpy many times before, so I had enough knowledge about it to use it comfortably in my project. However, I hadn't ventured into computer vision before the project aside from knowing very generally what a face descriptor is.

2.8.2 Tools Used

The project was written in Python 3.6. In order to separate the environment that the project needs with my local environment, I used a virtual environment to contain the installations of the required libraries. I also had a requirements.txt file to contain the names and versions of the packages that were used. I also used the following libraries:

1. **Numpy**: Using numpy helped increase performance of many parallel computations, such as matrix multiplication, and provided a simple interface to do these with.

2. **Keras and Tensorflow**: I used Keras to create the VGG model and the CifarNet model. We then use it to train the CifarNet model and then make predictions for both models. The weights of VGG were already pre-computed so no training was necessary for VGG.

3. **OpenCV**: I used OpenCV for general image processing, such as converting an image into grayscale or loading in images, and for face detection.

4. **Sklearn**: This library provided an implementation for K-Nearest Neighbors and PCA.

5. **Scipy**: This library provided a function that solved the general eigenvalue problem, given two numpy arrays which was used in the main WGEML algorithm.

Finally, Git and Github were used for version control and backups. Branches were created for each feature that was to be added to the project and I merged them into the master branch once enough testing was done, whether it was unit testing or integration testing.

2.8.3 Datasets

I used the KinFaceW-I, KinFaceW-II, and TSKinFace datasets to train the model.

The KinFaceW datasets [4] [5] are composed of face images from the internet which include public figures as well as their parents or children. Both datasets contain no restriction on pose, lighting background, expression, age, ethnicity, or partial occlusion. The main difference between the datasets is that the pairs of face images that have a kin relationship in KinFaceW-I are from different images whereas, in KinFaceW-II, they are from the same image. In terms of the specifics of the datasets, they both support four kin relationships, Father-Son (FS), Father-Daughter (FD), Mother-Son (MS), and Mother-Daughter (MD). Each face image are images of size 64×64. The datasets also pre-computed 5 folds for cross-validation for each relationship which contained the names of the pairs of images that had the relationship and those that didn't, henceforth positive and negative pairs respectively. Finally, the dataset has 2 main settings which are used: the restricted setting in which only positive pairs of images are used and the unrestricted setting in which negative pairs of images are used.

The TSKinFace dataset [8] contains images for the relationships, Father-Mother-Son (FMS), Father-Mother-Daughter (FMD), and Father-Mother-Son-Daughter (FMSD). The dataset contained folders for each relationship and a positive set comprised of the images in which the names of the images had the form "[relationship]-N-[member].jpg" in which "relationship" was the relationship, N was a consistent number and "member" refers to which member of the relationship they were. For example, "FMS-10-F.jpg", "FMS-10-M.jpg", and "FMS-10-S.jpg" would form a positive triplet. The dataset only contained the images in this form so negative pairs of images and the folds for cross-validation had to be computed in the project.

I examine the effects of the WGEML algorithm on the FS, FD, MS, MD, FMS and FMD relationships, as defined above.

2.8.4 Testing

Throughout the project, I created unit tests for any modules and for each function in the module. I would only end up merging a feature branch into the master branch if all of the unit tests passed for that feature.

Along with this, I did integration tests in the form of feeding the scripts small, controlled inputs to see if the scripts would output what was expected. These scripts each used functions from different modules and did a part of the workflow.

Finally, an end-to-end test was done once each script was tested individually which came in the form of trying the small, controlled input for the first script and then running it through each of the scripts successively.

2.8.5 Licensing

The SIFT algorithm had been patented in the US. However, the patent expired last year, and the patent was only to protect stop commercial use of the algorithm, whereas this is an academic use of the algorithm.

Furthermore, I am able to use VGG for non-commercial purposes under the Creative Commons Attribution License.

Chapter 3

Implementation

3.1 Repository Overview

3.2 Overview of Workflow

3.3 Face Descriptors

3.4 WGEML

3.5 Prediction

3.6 Data Preparation

3.6.1 Cross-Validation

3.6.2 Positive and Negative Pairs

3.6.3 PCA

3.6.4 Saving Results to Disk

3.7 CifarNet Extension

Chapter 4

Evaluation

4.1 Success Criterion

4.2 Original Results

4.3 Potential Biases in Datasets

4.4 Ablation Studies

4.4.1 Blocking Face Descriptors

4.4.2 Blocking Parts of Images

4.5 Replacing VGG

4.6 Unit Tests

Chapter 5

Conclusion

5.1 Achievements

5.2 Lessons Learnt

5.3 Future Work

Bibliography

[1] R. Fang, K. D. Tang, N. Snavely, and T. Chen. Towards computational models of kinship verification. In *2010 IEEE International Conference on Image Processing*, pages 1577–1580, 2010.

[2] W.D. Hamilton. The genetical evolution of social behaviour. ii. *Journal of Theoretical Biology*, 7(1):17 – 52, 1964.

[3] J. Liang, Q. Hu, C. Dang, and W. Zuo. Weighted graph embedding-based metric learning for kinship verification. *IEEE Transactions on Image Processing*, 28(3):1149–1162, 2019.

[4] J. Lu, J. Hu, X. Zhou, Y. Shang, Y. Tan, and G. Wang. Neighborhood repulsed metric learning for kinship verification. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2594–2601, 2012.

[5] J. Lu, X. Zhou, Y. Tan, Y. Shang, and J. Zhou. Neighborhood repulsed metric learning for kinship verification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(2):331–345, 2014.

[6] Jill M. Mateo. Perspectives: Hamilton's legacy: Mechanisms of kin recognition in humans. *Ethology*, 121(5):419–427, 2015.

[7] A. Nandy and S. S. Mondal. Kinship verification using deep siamese convolutional neural network. In *2019 14th IEEE International Conference on Automatic Face Gesture Recognition (FG 2019)*, pages 1–5, 2019.

[8] Xiaoqian Qin, Xiaoyang Tan, and Songcan Chen. Tri-subject kinship verification: Understanding the core of a family, 2015.

[9] Joseph P Robinson, Ming Shao, and Yun Fu. Visual kinship recognition: A decade in the making, 2020.

[10] Jun Yu, Mengyan Li, Xinlong Hao, and Guochen Xie. Deep fusion siamese network for automatic kinship verification. *2020 15th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2020)*, pages 892–899, 2020.

Appendix A

Project Proposal