

Computer Science Tripos

Part II Project Proposal Coversheet

Please fill in Part 1 of this form and attach it to the front of your Project Proposal.

Part 1

Name:

Manu Varma

CRSID:

mv465

College:

St. John's College

Overseers: (Initials)

FMS and ASP

Title of Project:

Kin Recognition using Weighted Graph Embeddings

Date of submission:

Will Human Participants be used?

No

Project Originator:

Manu Varma

Signature:

Manu Varma

Project Supervisor:

Daniel Bates

Signature:

Directors of
Studies:

Robert Mullins

Signature:

Special Resource
Sponsor:

Robert Mullins

Signature:

Special Resource
Sponsor:

Robert Mullins

Signature:

Above signatures to be obtained by the Student

Part 2

Overseer Signature 1: -----

Overseer Signature 2: -----

Overseers signatures to be obtained by Student Administration.

Overseers Notes:

Part 3

SA Date Received:

SA Signature Approved:

Kin Recognition using Weighted Graph Embeddings

Computer Science - Part II Project Proposal

Manu Varma

1 Introduction and Background

Kin recognition is the way of recognizing whether two people are related or not based on their faces and, if they are related, how are they related. It is advantageous to inclusive fitness for an organism to be able to recognize which of their neighbors were close relatives [4]. Thus, it stands to reason that the ability to recognize kin relationships has evolved in humans. In humans, facial resemblance is expected to serve as an indicator of kinship. Strangers are able to match photographs of mothers to their infants without any prior contact with the family [7].

In the field of creating computational models of kin recognition, there are multiple types of problems that are being solved. The first and most common kind is kinship verification which is what will be explored in this project. The task is to determine whether a pair of faces are blood relatives or not and what type of relatives they are. The next type of task is family classification which is the task of determining which family a single member belongs to. The next type of task is tri-subject verification where it is determined whether a child is related to a pair of parents. Finally, there is the search and retrieval task where a child is checked against all of the faces in a database to see if there is anyone they are related to. This has applications in helping find missing children specifically [8].

2 Starting Point

2.1 Personal Starting Point

Currently, I have minimal experience with Tensorflow and Keras. I have also worked with facial recognition a few years ago. I am very familiar with Python, overall, however. The main courses that could be helpful would be Machine Learning and Bayesian Inference and Deep Neural Networks in which I will be attending both in Lent Term and I will try to read ahead in before I start my implementation.

2.2 Current Existing Literature

One of the earliest approaches to kinship recognition [3] obtained the main facial features from each person using a pictorial structure model which then got the features and then a feature vector was created from that. Using the feature vectors, the differences between the corresponding feature vectors were calculated before applying K-nearest neighbors and Support Vector Machine methods to train the model.

Another method used in 2012 was to first partition the face into regions in 5 layers by slowly breaking down the face in each layer [9]. This analyzes the facial features and compares the features between two people. However, due to aging, these features can be deformed a bit so this is mitigated using transfer subspace learning. They do this transfer learning on child-old parent and child-young parent pairs in order to create a new problem. This way, the old and young parents can be operating on the same distribution of features rather than different distributions. This allows the feature difference that young and old parents might have to decrease and establishes a new standard for getting the features of the face from each person. After this, they use the gender relation, age difference and the distance between the two people in the photo along with the results of the transfer learning to get a kinship score and pick the relationship that has the highest score.

Lastly, a more recent method from 2019 uses Weighted Graph Embedding-Based Metric Learning [5] to obtain the classification. Instead of obtaining a single feature vector, four types of feature descriptors are obtained. These descriptors are feature vectors obtained from Local Binary Patterns, a Histogram of Gradients, a scale-invariant feature transform vector and a 4096-dimensional VGG-Face CNN descriptor using the VGG-Very-Deep-16 CNN architecture. Once these descriptors are obtained, for each type of relationship (father-son, mother-daughter, etc.), an intrinsic graph and two penalty graphs are created. K-nearest neighbors is used on these graphs to construct a set of matrices which are subsequently used to figure out whether the inputted people have the relationship the graph was testing. When the method was evaluated on KinFaceW-I and KinFaceW-II, the mean verification accuracy was about 78% and 83% respectively.

3 Substance and Structure of the Project

3.1 Aims of the Project

I aim to replicate the kin recognition results using the Weighted Graph Embedding Based method [5]. Once that is done, I aim to extend the model with the following extensions:

- Many current implementations of kin recognition use the fact that they faces came from the same image which makes kin recognition easier [1]. We can try to extend

the current implementation and create our own model for this problem to better account for this.

- The functionality can be extended to also include video rather than just images.
- Extend the functionality beyond kin recognition to relationship recognition to also identify if people are friends as well or if they are strangers
- Add on a system where you can search for all of the people in a database that you might be related to in general or to find pictures of a specific family member.

3.2 Dataset

I will be using a combination of the KinFaceW-I [6] and Family101 [2] datasets. The KinFaceW-I dataset contains pairs of faces collected from the internet which are labeled with one of four kin relations: Father-Son (F-S), Father-Daughter (F-D), Mother-Son (M-S), and Mother-Daughter (M-D). For each pair of faces, each face comes from a different photo than the other face in the pair. The photos are taken from uncontrolled environments where there was no restriction in the pose, lighting, background, expression, age, ethnicity, and partial occlusion which would mean that the dataset is not very biased.

The Family101 dataset contains 101 different families with 607 individuals and 14,816 images composed of renowned public families.

For the potential extension of including videos, a variant of the KinFaceW dataset can be used called Kinship Face Videos in the Wild [10]. This is similar to the KinFaceW-I dataset except the resolution of each frame is 900×500 pixels and each video contains 100-500 frames each.

All of these datasets are public so long as they are cited as done above. In addition to being cited, the video dataset requires the creators to be emailed as well and the Family101 dataset requires them to be informed about the accuracies obtained.

3.3 Structure of the Project

To achieve this goal, a similar method to what is used in the weighted graph embedding approach [5]. The main components are as follows:

- I will require face recognition first in order to get the person's face from a given image into the format that will be used by the rest of the system.
- From the face that is a standardized size, the necessary feature descriptors need to be taken from the face image.

- The next component is the intrinsic graph and penalty graphs that are created for each kin relationship.
- Finally, the needed matrices that are obtained from said graphs are created to figure out whether the pair of faces has the relationship that is being tested.

3.4 Evaluation

One method of evaluation is to use the training/testing data split that the datasets already provide and use that as a benchmark for accuracy. We can use the KinFaceW-I and II datasets to benchmark the implementation with the paper and the Family101 dataset to see how it extends to other data. We can use currently existing models as a benchmark and compare our accuracy that was obtained with existing accuracies. We can also create a confusion matrix and calculate the F_1 -score of the model based on the results of using the test data. Furthermore, the validation and learning curves can be analyzed in order to figure out if the model is underfitting or overfitting.

4 Success Criterion

The project can be considered a success if it can successfully replicate the accuracies that the Weighted Graph Embedding-Based methodology towards kin recognition [5] obtained in their evaluation with an error range of $\pm 15\%$ due to factors such as the training/testing split being different from that used, using a different random number seed and using a different framework than was used in the paper or if I can invalidate the results of the paper.

5 Timeline and Milestones

1. Oct 23 - Nov 06:

This time will be mainly used for research and getting familiar with everything I need to know about the subject such as:

- Reading up more in depth on existing literature
- Getting a more in depth idea of what architecture I should be using
- Figuring out which libraries are necessary to be used
- Getting an understanding of the libraries to be used by doing small quick projects in either Tensorflow or PyTorch.

2. Nov 06 - Nov 20:

Basic face recognition will be implemented so that generic images can be given and not just images of a specific size of just their face.

- Using Dlib or OpenCV, create basic facial recognition software that outputs the image of the person's face in the required format.

Deliverable: A basic, usable facial recognition model that can be used for the rest of the project.

3. Nov 20 - Dec 04:

Create the ability to get the necessary feature vectors like local binary patterns and the histogram of gradients.

- Given the input of the pictures of the faces, get the local binary patterns and histogram of gradients out from them.

4. Dec 04 - Dec 18:

The VGG-Face CNN descriptors and SIFT face descriptors will be obtained in this time which would then be fed into the graphs. If this is done before the sprint is up, work will be started on implementing the graphs.

Deliverable: The face descriptor methods are all created.

5. Dec 18 - Jan 01:

The implementation of the intrinsic graph would be created in this sprint. This should create the graph based on the class information.

6. Jan 01 - Jan 15:

The penalty graphs should be created in this sprint and thus get the calculations necessary to figure out the kin relationship. This overall model that is created should be able to fulfill the success criterion.

Deliverable: A model that can predict kin relationships between pairs of images.

7. Jan 15 - Jan 29:

At this point, ablation studies will be done to help to evaluate the network by seeing which inputs are necessary for the network to work with more focus being given to the progress report and presentation:

- Start work on an ablation study.
- Write up the Progress Report and create a presentation for it to be handed in on February 5th.

Deliverable: Progress Report

8. Jan 29 - Feb 12:

Finish work on the ablation studies and start work on the first extension:

- Finish up ablation studies for evaluation and finish model
- Start working on extending the model to work for videos as well.

Deliverable: A finalized model that reports the kin relation between a people in a pair of images.

9. Feb 12 - Feb 26:

Finish work on the first extension and start doing work on a second extension to try and improve on the model for images.

Deliverable: A new model which is the model implemented which is extended to work on videos

10. Feb 26 - Mar 12:

At this point, the first draft of the dissertation starts to be written alongside some work on improving the model, with priority given to the draft:

- Write out a draft of the Introduction and Preparation Chapters
- Work on second extension for improving the existing model.

11. Mar 12 - Mar 26:

Continue working on writing the dissertation and extension with the same priority:

- Write out a draft of the Implementation chapter
- Work on second extension for improving the existing model. If it is finished, then this will be included in the implementation chapter. If not, it will be abandoned in favor of finishing up the draft of the dissertation.

Deliverable: A draft of the Introduction, Preparation and Implementation chapters to be given to supervisor for feedback

12. Mar 26 - Apr 09:

Continue working on writing the dissertation:

- Write out a draft of the Evaluation chapter
- Revise the Introduction, Preparation and Implementation chapter based on supervisor feedback

13. Apr 09 - Apr 23:

Continue working on writing the dissertation:

- Write out a draft of the Conclusion chapter

Deliverable: A draft dissertation to be given to supervisor for feedback

14. Apr 23 - May 14:

If everything has gone well, this should be where final touches are applied.

Deliverable: The completed dissertation

6 Resources Declaration

I plan to use my current laptop which is a Windows laptop with an Intel i7 at 2.8GHz, 16GB RAM, an NVIDIA GTX 1060 graphics card, 1TB of HDD space and 256GB of SSD space. I accept full responsibility for this machine and I have made contingency plans to protect myself against hardware and/or software failure. Such contingencies include backing up my files to Github, Google Drive and an external 1TB hard drive. The Github repository will be updated fairly often as I work on my project and the Google Drive and hard drive will be updated every 1-2 weeks. Should my machine encounter software failures, I would buy a new laptop and work on the MCS machines until the new laptop arrives. Furthermore, I will utilize GPUs provided by the Computer Laboratory when I need to train any models.

References

- [1] Mitchell Dawson, Andrew Zisserman, and Christoffer Nellåker. From same photo: Cheating on visual kinship challenges. *CoRR*, abs/1809.06200, 2018.
- [2] R. Fang, A. C. Gallagher, T. Chen, and A. Loui. Kinship classification by modeling facial feature heredity. In *2013 IEEE International Conference on Image Processing*, pages 2983–2987, 2013.
- [3] 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.
- [4] W.D. Hamilton. The genetical evolution of social behaviour. ii. *Journal of Theoretical Biology*, 7(1):17 – 52, 1964.
- [5] 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.
- [6] 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.
- [7] Jill M. Mateo. Perspectives: Hamilton’s legacy: Mechanisms of kin recognition in humans. *Ethology*, 121(5):419–427, 2015.
- [8] Joseph P Robinson, Ming Shao, and Yun Fu. Visual kinship recognition: A decade in the making, 2020.

- [9] S. Xia, M. Shao, J. Luo, and Y. Fu. Understanding kin relationships in a photo. *IEEE Transactions on Multimedia*, 14(4):1046–1056, 2012.
- [10] Haibin Yan and Junlin Hu. Video-based kinship verification using distance metric learning. *Pattern Recogn.*, 75(C):15–24, March 2018.