Generative Aging of Photographs for Kinship Verification

Yash Chandramouli
Dept. of Aeronautics and Astronautics
Stanford University
yashc3@stanford.edu

Zoe Ghiron
Dept. of Aeronautics and Astronautics
Stanford University
zghiron@stanford.edu

***Abstract*—The task of kinship verification from photos, namely determining whether two people are related given input images of each person, has use cases in areas ranging from combating human trafficking to conducting non-invasive paternity tests. Recently, techniques leveraging deep learning have shown improvements in handling this task; however, despite recent advances in the field of face recognition, performance on the kinship verification task has begun to stagnate, perhaps indicating the presence of a high Bayes error for this task. This work demonstrates that by using state-of-the-art generative models to transform the input faces to the same age, the error on this task can be reduced for CNN-based kinship verification algorithms in a model-agnostic way.**

I. Introduction

Kinship verification is the task of determining whether two people are blood relatives. This task has a variety of uses including fighting human trafficking, verifying paternity, and improving our understanding of human genetics and inheritance. Currently kinship verification is performed using genetic tests that are expensive, long, and require physical DNA samples. Computational methods for kinship verification based on photographs aim to alleviate these problems while simultaneously allowing researchers to better understand the hereditary properties of particular facial features.

Past approaches for kinship verification from photographs have been limited by the availability of large and diverse data-sets. It can also often be difficult, given the limited data-set, to isolate which features of a photo are due to the effects of aging and which are biological markers that could aid in verifying kinship. With these two problems in mind, we propose a novel architecture to improve on the current state-of-the-art in kinship verification. The input to our algorithm is two images from the Families in the Wild (FIW) database. These images are then aged to be the same age by a generative model before being sent to a CNN that encodes the images and determines whether they belong to two relatives. This process aims to minimize the effects of age-related features and by coupling it with the FIW database, which is the largest kinship database to date, we believe we will be able to improve upon the state-of-the-art.

II. Related work

A. Work prior to FIW

Prior to the release of the FIW data-set, the state-of-the-art in kinship verification was primarily focused on tackling specific sub-problems, such as identifying Father-Daughter(FD) relationships. Zhang et al. [1] combined the techniques of metric learning with deep learning to identify non-linear features and were able to attain reasonable ($\approx 70\%$ accuracy) on the KVII dataset. However, this approach was limited to only a few "simple" relationships such as Father-Daughter, and the dataset neglected looking at relationships with more generational differences (Grandfather-Granddaughter) or non-direct hereditary connections (Siblings). Other work, most notably the work of Lu et al. [2], utilized metric learning to obtain high-quality kinship verification results in 2015. However this again did not take advantage of a more expansive dataset and also required a similarity metric that had to be heavily tuned to achieve proper performance.

B. After FIW

There is still a large amount of work to be done using the FIW data-set due to its recent release. However, within their original paper, Robinson et al. provided a simple approach to kinship verification using various CNNs and the FIW dataset [3]. They implemented a variety of algorithms for the image-unrestricted problem (where the IDs of the images are known) and found the SphereFace algorithm [4] [5] to perform the best. However this still only attained an overall accuracy of 69.18%. Other work on the dataset includes SelfKin, the winner of the first FIW challenge in 2018 [6]. They utilized a novel form of self-adjusted weights to improve the detection of facial features and were able to attain an accuracy of 68.20% averaged across all relationship categories for the image-restricted problem (where the IDs of the photos are unknown). These two results currently comprise the state-of-the-art in the field and indicate that perhaps the Bayes Error for the task of estimating kinship between two images of different ages is inherently high. This in turn motivated us to look into facial aging techniques to modify the inputs to the problem and potentially create an "easier" problem.

C. Aging for Kinship Verification

Lastly, just to further motivate our methodology it is important to note that past work has also been done using facial aging to improve the accuracy of kinship verification. [7] [8]. These papers demonstrated that converting the two photos to be the same age did yield improved classification performance. However, they both utilized style transfer to age the photos, which required the presence of an intermediate data-set of "young photos" of the parents. This is therefore not applicable to data-sets without multiple images of the parents or in situations where the relationships are closer in age and therefore such intermediate photos are not available, such as Brother-Sister. By using FIW and more advanced aging algorithms that incorporate GANs, we aim to take this same concept but apply it with fewer constraints to improve on the state-of-the-art.

III. Dataset and Features

Families in the Wild (FIW) is an open-source database released by Robinson et al in November 2018, designed specifically for the tasks of kinship verification and classification [9]. The database contains more than 10,000 family photos from more than 1,000 families.

A selection of 163 different families was generated from [9] due to some of the families in the dataset not being publicly available. This dataset contains 50,000 pairs of 224x224 RGB images with each pair consisting of two images, some from the same family and some not. The dataset was split into training/dev/test sets with the following proportions:

- Training set: 153 families,
- Dev set: 5 families
- Test set: 5 families

Because the training will utilize the triplet loss, the training set is composed of triplets (anchor, positive and negative) while the dev and tests sets are composed of pairs and a label of "kin" or "not kin". Examples of these are shown in figures 1 and 2. All pictures are normalized by 255.

Overall, the training set contains 46,735 triplets, the dev set contains 3,276 pairs, and the test set contains 1,558 pairs.

(a) Anchor (b) Positive (c) Negative

Fig. 1: One example of the training set

IV. Methods

A. CNN

The core of this implementation is the convolutional neural network (CNN). A CNN is a specific form of neural network that utilizes mathematical operations known as convolutions.

(a) Picture 1 (b) Picture 2

Fig. 2: One example of the dev set with label = 1

These operations allow the network to take in high-dimensional inputs (such as images) and process them more efficiently than a standard neural network would by leveraging shared parameters between parts of the image as well as utilizing more sparsely-connected layers. For this paper, we focused on using the CNN to generate encodings for each image and then comparing the image encodings against a threshold ϵ in order to assess kinship probability.

1) A note on Transfer Learning: Since the focus of the paper was originally to demonstrate state-of-the-art kinship verification performance, the preliminary plan was to utilize transfer learning on the SphereFace algorithm [4] and fine-tune the later layers using the FIW dataset since this was the implementation used by the authors of the FIW paper. However, due to the lack of documentation on their specific implementation, there was difficulty in replicating the results from that paper and achieving similar levels of dev set accuracy. As a result, it was decided to use a simplified form of the SphereFace algorithm and build it from the ground up for the purpose of assessing the relative benefit of the GAN. Future work will involve fully implementing the SphereFace algorithm with the GAN front-end in order to hopefully attain state-of-the-art performance on this task.

2) The triplet loss: The loss function was chosen to be a simple triplet loss that minimizes the distance between an anchor image and an image from within the anchor's family while maximizing the distance between the anchor and an image from outside the anchor's family. More specifically:

- The CNN 's output is a vector of 128 elements **encoding** the input.
- Each example of the training set is a triplet (three different images): an anchor encoded as A, a positive encoded as P (belonging to the anchor's family) and a negative encoded as N (not belonging to the anchor's family).
- The triplet loss is : $L(A, P, N) = \max(||A-P||_2^2 - ||A-N||_2^2 + \alpha,\ 0)$ where α is a hyperparameter.
- The dev and tests sets consist of two pictures (encoded as $P1$ and $P2$) and one label (the label is 1 if the people belong to the same family and 0 otherwise). The distance between the encodings $P1$ and $P2$ is computed. If it is sufficiently low i.e. $||enc(P1) - enc(P2)||_2^2 < \epsilon$, $P1$ and $P2$ are predicted to belong to the same family. This introduces another hyperparameter ϵ.

3) Detailed presentation of the CNN: Using this loss function, a simple CNN inspired by [4] was trained to generate an encoding given an input image:

- 64 filters (3,3) with stride 2 and valid padding. RELU activation function,
- 128 filters (3,3) with stride 2 and valid padding. RELU activation function,
- 256 filters (3,3) with stride 2 and valid padding. RELU activation function,
- 512 filters (3,3) with stride 2 and valid padding. RELU activation function,
- Fully connected layer towards a 512-element output vector,
- Fully connected layer towards a 128-element output vector. This vector is normalized. This vector is the encoding of the input

Once the encodings were computed, it is then necessary to compute the best threshold ϵ. This is a value such that if the L2 distance between an image's encoding and the encoding of another image in its family is below ϵ, the model predicts that they are related. ϵ was chosen to maximize the accuracy on the dev set. If E_1 is the encoding of the first image in a pair then:

$$D = |||E_1 - E_2|||^2, D < \epsilon \Rightarrow same family$$

B. Face-Aging Algorithm

The second part of our implementation was a Generative Adversarial Network (GAN). This is a network trained in "competition" against a discriminator until it is able to fool the discriminator and produce images with a high enough quality. At the end of training, the discriminator should be randomly guessing as to whether its inputs were generated by the GAN or come from real data. Two state-of-the-art generative algorithms for face aging were trained and used to age the faces for this project. The details on both are below:

1) CAAE: One of the models used was the Conditional Adversarial Auto-Encoder (CAAE) [10] which can be found at [11]. This network uses an encoder (E) to encode the input image, which is then fed into a generator (G) that aims to create an aged face by propagating certain features of the image. Two discriminators labelled D_z and D_img act to validate the encoding and the output image respectively and train both. The result is a network that is capable of both aging forward (progression) and backward (regression). One downside is that it has a normalizing effect on the features of the face, which can sometimes remove identifying features of the image.

2) IPCGAN: The second algorithm used was the Identity-Preserved Conditional GAN (IPCGAN) [12] which can be found at [13]. A conditional GAN (c-GAN) is a GAN architecture that generates images that meet a certain criteria. This algorithm used age as the condition and also imposed a loss function dependent on whether the identity of the person in the image was preserved as well as whether an age classifier predicted the correct age from the generated image. This algorithm achieved high performance on a dataset of celebrity images but the downsides are that it can only age forward in time and the dataset it was trained on primarily contained photos of adults.

V. Experiments/Results/Discussion

A. Training of the CNN

In the original paper, the benchmark using a full implementation of the SphereFace CNN achieved only a 69.13% test set accuracy. This coupled with the fact that the maximum achieved training set error for our task was only 84% indicates that this is likely a hard problem with a high Bayes error. This makes reasonable sense since it is also a task that is very difficult for humans to do with 70% accuracy. Therefore, the algorithm was considered to be sufficiently trained if it achieved approximately 70% training accuracy.

1) CNN Hyperparameters:

a) The learning rate: The learning rate was chosen to be $1e^{-5}$. Every 5 epochs, it is divided by 5. This was empirically chosen to prevent overshooting and to ensure that the cost keeps decreasing.

b) The threshold ϵ: This was specific to the triplet loss and was re-computed after each simulation in order to maximize accuracy on the dev set.

c) The triplet loss hyperparameter α: This parameter needs to be more than 0. It was empirically selected at $\alpha = 0.4$ after multiple tests.

d) Number of epochs: 5, 10, 20 and 30 epochs were all attempted. As shown on Table I, for a training set of 20000 examples (mini-batch size of 128), the test accuracy after 20 epochs is 54.8 % (training accuracy is 81 %) whereas it is 57.7 % after 5 epochs (training accuracy is 68 %). In other words, when the number of epochs is increased beyond 5, the training accuracy increases and the test accuracy decreases.This indicates overfitting. Note that the dev set accuracy is approximately constant because the hyperparamater ϵ is chosen to maximize accuracy on the dev set. Overfitting was avoided using the early stopping method and only training up to 5 epochs. The resulting 57.7% test accuracy is acceptable since similar simplistic algorithms in the original FIW paper also achieved 55-60% test accuracy.

Number of epochs	Dev set accuracy [%]	Training set accuracy[%]	Test set accuracy [%]
30	54	84	56.4
20	56.1	80.7	54.8
10	55.4	76	56
5	53.6	67.9	57.7

TABLE I: Early stopping

e) Mini-batch size: Mini-batches of size 32, 64 and 128 were tested on 20000 training examples and with 5 epochs. The results are shown in the Table II. If the mini-batch size was higher than 256, the computation was too slow: with numpy matrixes of dimensions (256, 224, 224, 3) almost throwing a

memory error. Since each picture is present several times in the training set, the result is that updating the parameters more often (with a small mini-batch) could increase overfitting: this explains the results in Table II. Thus, the mini-batch size was chosen to be 128 to avoid overfitting. The results presented in the Table II are for a training of 5 epochs on 20000 examples.

Mini-batch size	Dev set accuracy [%]	Training set accuracy[%]	Test set accuracy [%]
32	52.8	75	53.8
64	55.1	75	55.1
128	53.6	67.9	57.7

TABLE II: Mini-batch search

2) Results of the CNN: The final hyperparameters used for training with 20000 examples are shown in Table III.

Hyperparameter	Value
Learning rate	$1E-5$
Threshold	1.19
α	0.4
Number of epochs	5
Mini-batch size	128

TABLE III: Hyperparameters

For the hyperparameters presented in Table III, the accuracy was 53.6% on the dev set, 67.9% on the training set and 57.7% on the test set. On the test set, the precision is 60.1% and the recall is 48.9% The confusion matrix is presented in Table IV.

		Predicted [%]	
		True	False
Labeled [%]	True	24.8	25.9
	False	16.4	32.9

TABLE IV: Confusion matrix

B. Training the Generative Networks

1) CAAE: In implementing the CAAE, first the weights were initialized using a very early checkpoint model and then were trained using 23,708 images from the UTKFace dataset [14]. This is a high-quality dataset of images across ages which makes it ideal for the purpose of training a face-aging network. This would also allow us to be able to use the FIW dataset as our "test set" to assess the quality of the final aging. The training happened using AWS on a g3.4xlarge instance and took around 6:40:12 to run with an epoch size of 20, and a mini-batch size of 100. The learning rate was chosen to be 0.0002 and the β_1 value for the Adam optimizer was set at 0.5, both according to the original paper. The one hyperparameter that had to be tuned most heavily was mini-batch size. Due to memory constraints, mini-batches greater than 500 were not possible. However, too low batch size values took too long to run. Ultimately the best results were achieved using a mini-batch size of 100. Figure 4 below shows the final results of training.

(a) Input Image to CAAE

(b) Output image from CAAE

Fig. 3: Input and Output images to CAAE with 20 epochs of training. Each column represents a different person while each row represents a different age group, ranging from 0 to 70 years old in increments of 7 years.

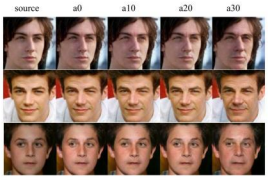

Fig. 4: Outputs from IPCGAN

2) IPCGAN: Due to time constraints, the IPCGAN was initialized with checkpoint weights provided by the authors of the paper which were then used to convert the images from the test set. The only hyperparameter selected was the number of age groups which was eventually chosen to be 5 to produce the best aged outputs. An example of these outputs can be seen in Figure 4.

C. Merging the Model

With the model trained using the parameters from section A above, the test set was run again but with the data pre-processed by the GANs such that the two people in each pair of test images were generatively aged into the same "age category". For the CAAE, each age category represents approximately a 7 year range from age 0 to 70. For the IPCGAN each age category represents approximately a 10 year age increase from the original age. The results for each GAN and each "age category" are presented in Table V.

Table V indicates that aging the photos does have a positive effect on the accuracy of the Kinship Verification algorithm. The best accuracy is achieved by aging the images down to Category 2 using the CAAE, which corresponds to what is expected. Without pre-processing the algorithm achieved a test

Pre-processing	Age category	Age [years]	Accuracy [%]	Precision [%]	Recall [%]
None	-	-	57.7	60.1	48.9
CAAE	1	0-7	56.7	56.5	**63.7**
	2	8-14	**60.4**	60.7	62.1
	3	15-21	58.8	59.6	58
	4	22-28	60.2	61.6	57.1
	5	29-35	59	60.9	53.2
	6	36-42	58.8	60.6	53.6
	7	43-49	58.7	61.6	49.1
	8	50-56	56.3	58.6	46.9
	9	57-63	56.5	58.8	47.3
	10	64-70	55.3	57.7	44.1
IPCGAN	1	+0	57.1	59.3	48.9
	2	+10	57.1	59.8	46.9
	3	+20	57.1	60.9	43
	4	+30	56.7	60.7	41.4
	5	+40	57.6	**63.1**	39.4

TABLE V: Results with Preprocessing

accuracy of 57.7%, a precision of 60.1% and recall of 48.9%. With pre-processing, it is able to obtain an accuracy of 60.4% (CAAE GAN - 2nd age category), a precision of 63.1% (IPC GAN - 5th age category) and recall of 63.7% (CAAE GAN - 1st age category).

It is also of note that when the model is made to overfit the non-processed data, while the performance on the non-processed test set suffers, the performance on data aged with the CAAE GAN achieves higher accuracy and precision. Choosing the same parameters as in Table III but using a mini-batch size of 64 and a threshold $\epsilon = 0.74$ causes the model to overfit and results in a a dev set accuracy of 55.1%, a training set accuracy of 75%, a test set accuracy of 55.1% without pre-processing. However, once the test data is preprocessed through the CAAE and set to age category 2, the accuracy is increased by 7% (up to 62.5 %) and the recall by 16% (up to 47%). For the 5th age category, the precision is increased by 9% (up to 69.7 %). These results can be viewed in Table VI.

An explanation for this could be that when the model overfits, it decreases the bias of the model but also hurts the variance. However, since the CAAE has a smoothing effect of sorts it means the processed test set inherently has less variance (i.e. encodes more "basic" features) than the original test set. This means the effect of decreasing the overall bias of the model outweighs the effect of the lower generalizability of the model caused by overfitting. It is important to note that the recall is still relatively low, indicating that the model fails to classify many true pairs as kin. This is likely because the encodings are overfitting the training set too heavily which in turn means fewer unprocessed test set pairs are meeting the "standard of similarity" learned by the model. However, like described above, the smoothing effect of the CAAE likely leads to more simplistic encodings that have fewer outlier features which in turn causes a large jump in recall performance when the images are pre-processed.

Pre-processing	Age category	Accuracy [%]	Precision [%]	Recall [%]
None	-	55.1	61.1	31.3
CAAE	1	57.3	60	47
	2	**62.5**	68.9	**47**
	3	60.7	66.5	45.3
	4	61	67.6	44.3
	5	61	**69.7**	40.8
	6	61	69.4	41.2
	7	59.2	67.2	38.0
	8	59.6	68.3	37.9
	9	57.8	64.6	37.0
	10	56	62.4	33.1

TABLE VI: CAAE pre-processed data applied to an overfitted model

VI. Conclusion/Future Work

Overall, from Tables 5 and 6 it can be seen that the generative algorithms add between 3-5% to test set accuracy and therefore confirm the hypothesis. Specifically, the highest accuracy is achieved by the CAAE scaling the images down to Age Category 2 (approximately between 7 and 14 years old). This is in line with our hypothesis since this increase in accuracy is likely due to age-related features being "smoothed out" by making the images younger. The drop in accuracy between Category 2 and Category 1 is likely because at Category 1, the features are so heavily smoothed by the CAAE that the encodings are inherently similar. This leads to a large amount of false positives as indicated by the low precision for that run. However, CAAE Category 1 did maximize recall due to this high encoding similarity which may be useful in situations where one may hope to avoid false negatives (such as paternity tests).

One surprising result is that the highest precision was actually achieved through aging forward using the IPCGAN and Age Category 5. This is likely because the IPCGAN preserves a large amount of individually-distinct features that inherently makes the encodings more different. Therefore, while many "harder" family pairs may be marked as false negatives, it ensures that those classified as "kin" are more likely to be so. This would be more useful for tasks, such as human trafficking detection, where false positives should be minimized.

One area of future work is to implement this pre-processing on the full SphereFace model and try to achieve state-of-the-art accuracy through this method. Another area of future work would be to incorporate an age estimator in the loop. This would allow for more interesting aging policies that could be explored, such as aging both images in a pair to the average age of the pair, or aging both images to the age of the younger image. Lastly, it would also be interesting to look further into formalizing a way to overfit the CNN so that it performs better on the pre-processed dataset. This is no longer a model-agnostic method, but it could be useful for improving model performance in specialized applications.

VII. Contributions

Zoe Ghiron worked on training the CNN, which included building it, implementing it, and conducting the hyperparameter search. She also wrote the helper functions to extract data from FIW and she generated the final results using the generatively-aged photos.

Yash Chandramouli worked on training the CAAE network and implementing the IPCGAN. The CAAE was trained without initial weights and required hyperparameter tuning.Both generative networks also required multiple modifications to core functions as well as multiple auxiliary functions in order to meet the needs of the project.

Our Github with the code for this project can be found at **https://github.com/yashc95/old-photo-net.git** [15].

References

[1] Kaihao Zhang, Yongzhen Huang, Chunfeng Song, Hong Wu, and Liang Wang. Kinship verification with deep convolutional neural networks. In Mark W. Jones Xianghua Xie and Gary K. L. Tam, editors, *Proceedings of the British Machine Vision Conference (BMVC)*, pages 148.1–148.12. BMVA Press, September 2015.

[2] Jiwen Lu, Xiuzhuang Zhou, Yap-Pen Tan, Yuanyuan Shang, and Jie Zhou. Neighborhood repulsed metric learning for kinship verification. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 331–345, July 2015.

[3] Joseph P Robinson, Ming Shao, Yue Wu, and Yun Fu. Families in the wild (fiw): Large-scale kinship image database and benchmarks. In *IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 40, No.11*. IEEE, November 2018.

[4] https://github.com/wy1iu/sphereface.

[5] Weiyang Liu, Yandong Wen, Zhiding Yu, Ming Li, Bhiksha Raj, and Le Song. Sphereface: Deep hypersphere embedding for face recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[6] Eran Dahan and Yosi Keller. Selfkin: Self adjusted deep model for kinship verification. *CoRR*, abs/1809.08493, 2018.

[7] Siyu Xia, Ming Shao, Jiebo Luo, and Yun Fu. Understanding kin relationships in a photo. In *IEEE Transactions on Multimedia, Vol. 14, No. 4*, 2012.

[8] Siyu Xia, Ming Shao, and Yun Fu. Kinship verification through transfer learning. In *22nd International Joint Conference on Artificial Intelligence*, 2011.

[9] https://web.northeastern.edu/smilelab/fiw/.

[10] Zhifei Zhang, Yang Song, and Hairong Qi. Age progression/regression by conditional adversarial autoencoder. In *IEEE Conference on Computer Vision and Pattern Recognition*. University of Tennessee, Knoxville, 2017.

[11] https://github.com/ZZUTK/Face-Aging-CAAE.

[12] Zongwei Wang, Xu Tang, Weixin Luo, and Shenghua Gao. Face aging with identity-preserved conditional generative adversarial networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[13] https://github.com/dawei6875797/Face-Aging-with-Identity-Preserved-Conditional-Generative-Adversarial-Networks.

[14] https://drive.google.com/drive/folders/0BxYys69jI14kU0I1YUQyY1ZDRUE.

[15] https://github.com/yashc95/old-photo-net.git.