Kinship Verification using Deep Siamese Convolutional Neural Network

Abhilash Nandy[1] and Shanka Subhra Mondal[1]
[1] Department of Electronics and Electrical Communication Engineering, Indian Institute of Technology, Kharagpur, India

***Abstract*— Recognizing Families In the Wild (RFIW) is a large-scale kinship recognition challenge based on the FIW dataset. This dataset is the largest databases for kinship recognition, consisting of more than 13,000 family photos and 1,000 families. The number of members in each family range from 4 to 38. One of the tasks for the database is, given photos of two individuals, predict whether they have any kin relationship or not. In this paper, we present a deep learning approach using Siamese Convolutional Neural Network Architecture to quantify the similarity between two given photos. We use two parallel SqueezeNet Networks, initialized with weights obtained after training the SqueezeNet on the VGGFace2 Dataset, and use a similarity metric and fully connected networks to merge the two networks to a single output. We use different similarity metric such as L1 norm, L2 Norm, and Cosine Similarity. Our network gives good accuracy and AUC scores.**

I. INTRODUCTION

For kinship verification, the Families In Wild (FIW) database [9], [10], [12] provides 11 types of relationship pairs - brother-brother(B-B), sister-sister(S-S), brother-sister(SIBS), mother-son(M-S), mother-daughter(M-D), father-son(F-S), father-daughter(F-D), grandmother-grandson(GM-GS), grandmother-granddaughter(GM-GD), grandfather-grandson(GF-GS), grandfather-granddaughter(GF-GD). Thus, it contains relationships across three generations. In order to tackle such a challenging problem, we use a siamese convolutional neural network architecture. The convolutional neural network architecture used is the SqueezeNet, and with weights learned from pre-training on the VGGFace2 Dataset [8], and then further training this architecture on the FIW dataset. The weights are initialized in such a manner so as to help in extracting the facial features from the data easily, thus leading to faster convergence.

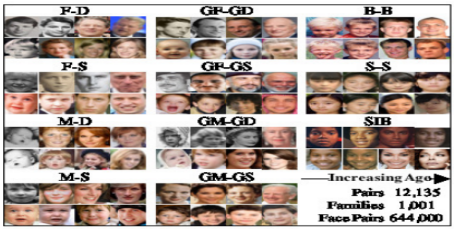
A. Related Works

Many approaches have been used to perform Kinship verification. Before the FIW dataset came into being, many studies on smaller Kinship datasets such as FamilyFace [13] and UB KinFace [14] have been performed. However, since FIW dataset is by far the largest of the data sets we test on, many experiments have been performed on the same in recent years. For instance, [1] used features extracted from two VGG Face Networks, concatenated them, and further used convolutional filters to extract high-level features. [2] used an ensemble of multiple deep nets for kinship verification. [3] fine-tuned a model used for face recognition on the FIW dataset using soft triplet loss function for backpropagation. Further, some baselines of [4] used VGG-Face with a triplet loss on top. [11] used a denoising auto-encoder based robust metric learning (DML) framework in order to learn features that not only maintained the inherent pattern of the data, but also proved to be capable of providing discriminating knowledge. [6] used non-neural methods of feature extraction - local phase quantization (LPQ) followed by Subspace reduction and classification on a pair of images. From this, cosine similarity was used to determine whether the two faces in the images share a relationship or not.

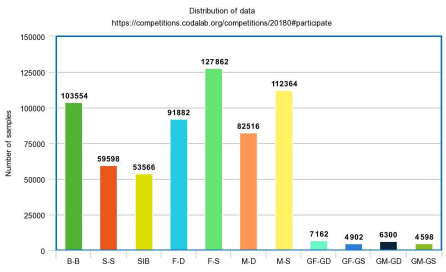
This paper is organized as follows. Section II gives the description of the FIW dataset, Section III provides an overview of the method used, Section IV shows the experiments and results, Section V is the conclusion.

II. DATASET DESCRIPTION

As already mentioned, in I, there are 11 types of relationship pairs available in FIW, as shown in Fig. 1a. The total data is very imbalanced, as can be seen by the distribution of the data in Fig. 1b. Relationships having a gap of two generations (for instance, GF-GS) have fewer examples as compared to relationships (for instance, F-S) having one generation gap, making the data diverse and posing a challenge against the problem of kinship verification.



(a)



(b)

Fig. 1: (a) FIW Faces and Counts For Kinship Types [5]. (b) Distribution of FIW data

978-1-7281-0089-0/19/$31.00 ©2019 IEEE

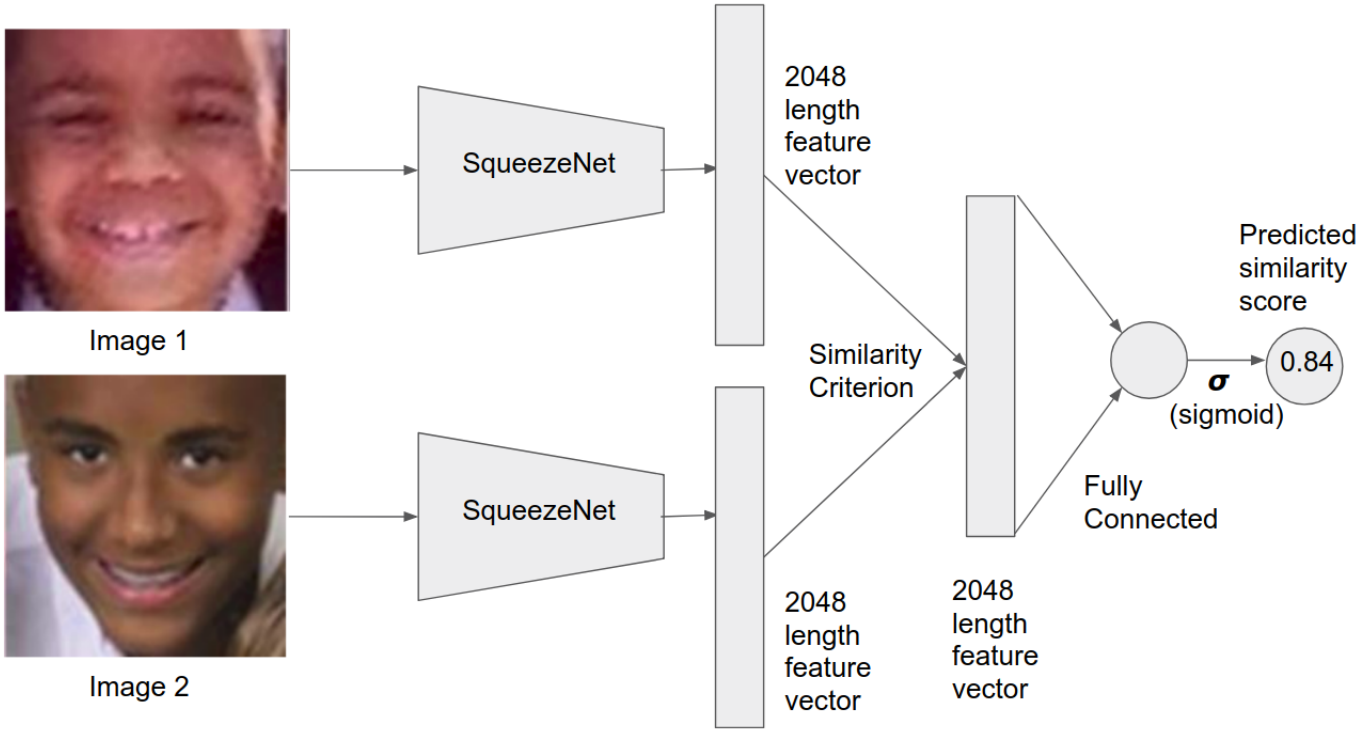


Fig. 2: Proposed Architecture for the Siamese Convolutional Neural Network. The convolutional neural network has a SqueezeNet architecture. Two convolutional branches are merged together at the top using a similarity criterion, which upon further processing finally gives the similarity score as the output.

III. Method Overview

We propose a simple Siamese convolutional neural network architecture, and experiment using different similarity metrics - L1 Norm, L2 Norm, and Cosine Similarity, with a simple binary cross-entropy loss function used at the end.

A. Data Augmentation

We use random horizontal flips and random rotations on the image. For faster convergence, we normalise the images (0, 1), and to apply pre-trained weights of the SqueezeNet Model, we do a channel-wise normalization, with a mean = $[0.485, 0.456, 0.406]$ and standard deviation = $[0.229, 0.224, 0.225]$, for the red, green, and blue channels, respectively[1].

B. Proposed Architecture

Our architecture, as shown in Fig. 2, consists of two SqueezeNet models (a SqueezeNet Model architecture is shown in Fig. 4) as the two branches of the siamese net, each of which gives a 2,048 length vector as the output. The reason behind using a SqueezeNet Model is that the Squeeze-and-Excitation (SE) blocks recalibrate feature responses in each of the channels adaptively by explicitly modelling relationships between channels. These blocks contain a 'squeeze' part, which consists of only 1x1 convolutional filters, while the 'excite' (or 'expand') part consists of 1x1 and 3x3 convolutional layers (see Fig. 5). Thus, SqueezeNet is able to extract features that account for not only intra-channel pixel feature relationships, but also, inter-channel relationships among the red, green and blue channels thus, enhancing the quality of features extracted. Another advantage of SqueezeNet is that, it has proven to have good performance in the ImageNet Challenge [7], with few number of parameters, which gives an intuition that it will not overfit the training data. To find the similarity between the two sets of extracted features, we apply a similarity metric on the two vectors to get one 2,048 length vector. A similarity metric is applied to merge the two branches of the siamese net thus, giving an output feature vector that encodes the similarity between the two input images. This is fully connected to the output neuron, followed by a sigmoid activation function that squeezes the output between 0 and 1, and is mapped to the target label of either having a relation (1) or not (0). Upon sufficient training, the predicted output would quantify the amount of similarity between the two inputs, in other words, the similarity score between the two inputs. The loss function used is a binary-cross entropy function, which is given as follows:

$$Loss = -ylog(p) - (1-y)log(1-p) \tag{1}$$

where y is the target value that is either 0 or 1, and p is the predicted value.

[1]https://pytorch.org/docs/stable/torchvision/models.html

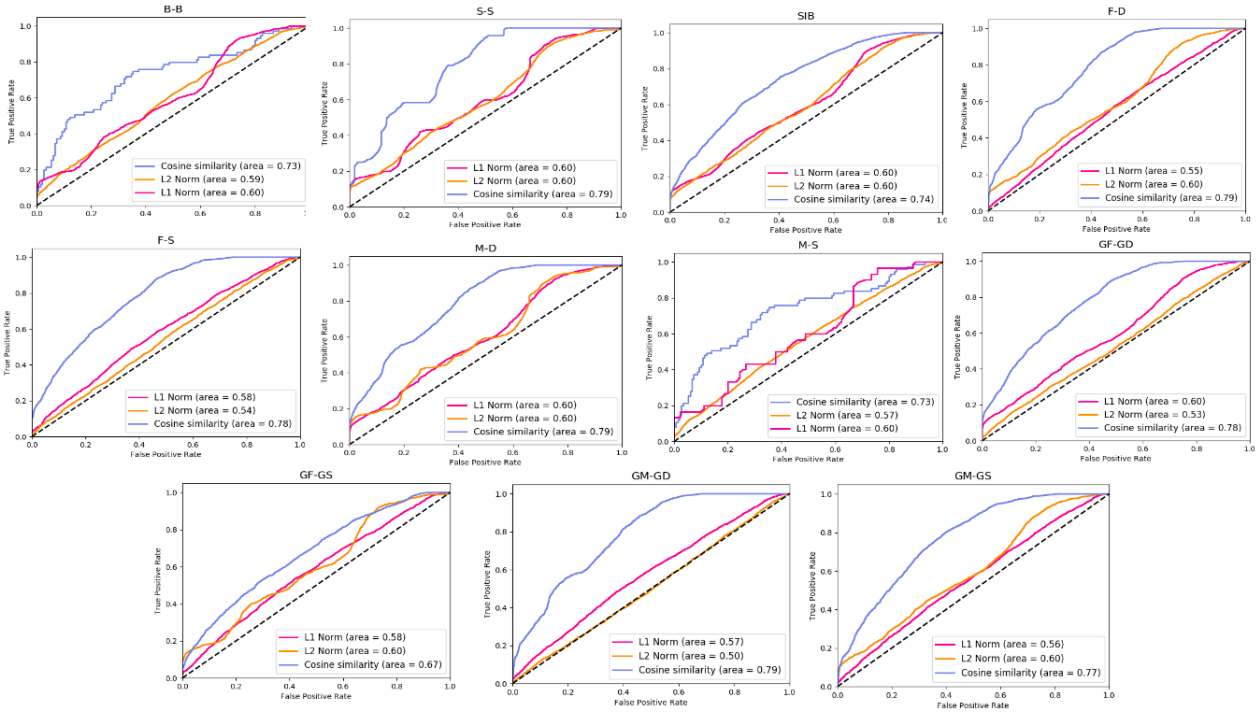


Fig. 3: Class-wise roc curves along with AUC scores for all the similarity criteria

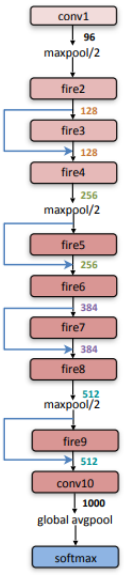


Fig. 4: SqueezeNet Architecture; here each of the 'fire' models has a 'squeeze' conv block, having 1x1 filters, followed by a 'expand' block, consisting of 1x1 and 3x3 filters. [7]

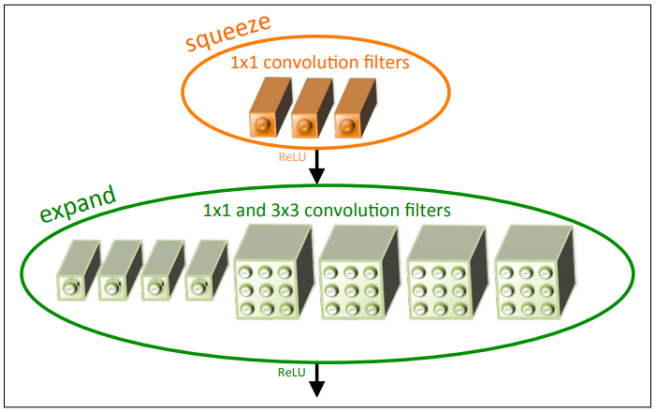


Fig. 5: Squeeze-and-excitation blocks [7]

C. Similarity Metrics used

For two vectors, $X = \{x_1, x_2, x_3, ..., x_n\}$ and $Y = \{y_1, y_2, y_3, ..., y_n\}$, we used the following similarity metrics:

1) L1 Norm:

$$||X - Y||_1 = \sum_{i=1}^{n} |x_i - y_i| \tag{2}$$

2) L2 Norm:

$$||X - Y||_2 = (\sum_{i=1}^{n} |x_i - y_i|^2)^{1/2} \tag{3}$$

3) Cosine Similarity:

$$K(X, Y) = \frac{X.Y}{||X||.||Y||} \tag{4}$$

In our case, we use these similarity metrics on each element of the two feature vectors at the end of the two convolutional branches, considering each element to be a vector of size 1 (for instance, we first find the similarity metric between the first elements of the two vectors, followed by the second element, and so on). Hence, the vector obtained after applying the similarity metric is of the same size as that of the two feature vectors.

IV. Experiments and Results

A. Training conditions and Hyperparameters Used

We used an Adam Optimizer with a learning rate of 0.001, beta1=0.9, beta2=0.99. We used a batch size of 64, and trained the network for over 40 epochs. To force both the branches of the network to have all the images as inputs, and not make one side biased to a single set of images, and the other side vice versa, every alternate epoch, we interchange the two images during training. Further, each SqueezeNet branch is initialized using weights of pretrained SqueezeNet architecture on VGGFace2 [8] dataset.

B. Results

Two evaluation metrics are employed in order to display results, Test accuracy and AUC scores.

1) Test Accuracy: The table I compares the performance in terms of test accuracy between various methods using different similarity metrics. From the table, it can be seen that cosine similarity performs the best of the all the three similarity metrics, giving about a 4% - 5% jump in the average test accuracy, as compared to the L1 and the L2 norms.

TABLE I: Comparison of test accuracies between different similarity metrics

Relation	L1 Norm	L2 Norm	Cosine Similarity
B-B	58.32%	59.21%	**64.23%**
S-S	62.64%	61.43%	**67.43%**
SIB	65.72%	63.84%	**67.85%**
F-D	58.87%	58.92%	**62.53%**
F-S	64.84%	63.24%	**68.74%**
M-D	63.25%	64.43%	**69.84%**
M-S	63.54%	62.64%	**67.95%**
GF-GD	66.44%	65.43%	**70.82%**
GF-GS	64.96%	64.32%	**68.42%**
GM-GD	62.43%	64.45%	**69.74%**
GM-GS	60.92%	62.42%	**66.74%**
Average	62.9%	62.75%	**67.66%**

2) AUC scores: We plot the ROC curves for each of the 11 classes, and for all the similarity criteria, as shown in Fig. 3, and tabulate the results in table II. We again see that cosine similarity performs the best of all the other similarity criteria for all the relationships.

TABLE II: Comparison of AUC scores between different similarity metrics

Relation	L1 Norm	L2 Norm	Cosine Similarity
B-B	0.60	0.59	**0.73**
S-S	0.60	0.60	**0.79**
SIB	0.60	0.60	**0.74**
F-D	0.55	0.60	**0.79**
F-S	0.58	0.54	**0.78**
M-D	0.60	0.60	**0.79**
M-S	0.60	0.57	**0.73**
GF-GD	0.60	0.53	**0.78**
GF-GS	0.58	0.60	**0.67**
GM-GD	0.57	0.50	**0.79**
GM-GS	0.56	0.60	**0.77**
Average	0.58	0.58	**0.76**

V. CONCLUSION

We propose fine-tuning of pretrained parallel SqueezeNet Networks and a similarity metric to combine the results from each part to finally predict kinship relation. The cosine similarity metric performs better than L1 and L2 metric with respect to accuracy because of effective and simple learning of the objective function. L1 and L2 norm restrict the data to be related in a linear and a quadratic fashion respectively which may not be the case, whereas, cosine similarity generalizes well, capturing relationships in transformed space. The squeeze and the excitation blocks in SqueezeNet model channel-wise relationships which help it to recalibrate channel-wise feature responses. Hence, the SqueezeNet architecture used in each of the branches of the Siamese Network helps in learning discriminating features from the raw facial image data.

VI. ACKNOWLEDGMENTS

The authors gratefully acknowledge the contribution of reviewers' comments. We would also like to thank Prof. Debdoot Sheet, Electrical dept., Indian Institute of Technology, Kharagpur, who suggested us to use various similarity metrics in order to do evaluation.

References

[1] Dahan, Eran, Yosi Keller, and Shahar Mahpod. "Kin-Verification Model on FIW Dataset Using Multi-Set Learning and Local Features." Proceedings of the 2017 Workshop on Recognizing Families In the Wild. ACM, 2017.

[2] Duan, Qingyan, and Lei Zhang. "AdvNet: Adversarial Contrastive Residual Net for 1 Million Kinship Recognition." Proceedings of the 2017 Workshop on Recognizing Families In the Wild. ACM, 2017.

[3] Li, Yong, Jiabei Zeng, Jie Zhang, Anbo Dai, Meina Kan, Shiguang Shan, and Xilin Chen. "KinNet: Fine-to-Coarse Deep Metric Learning for Kinship Verification." In Proceedings of the 2017 Workshop on Recognizing Families In the Wild, pp. 13-20. ACM, 2017.

[4] Robinson, Joseph P., et al. "Recognizing families in the wild (rfiw)." Proceedings of the 2017 ACM Workshop on RFIW17. 2017.

[5] Joseph P Robinson, Ming Shao, Yue Wu, and Yun Fu. 2016. Families in the Wild (FIW): Large-Scale Kinship Image Database and Benchmarks. In Proceedings of the 2016 ACM on Multimedia Conference. ACM, 242246.

[6] Laiadi, Oualid, Abdelmalik Ouamane, A. Benakcha and Abdelmalik Taleb-Ahmed. RFIW 2017: LPQ-SIEDA for Large Scale Kinship Verification. RFIW '17 (2017).

[7] Iandola, F. N., Han, S., Moskewicz, M. W., Ashraf, K., Dally, W. J., & Keutzer, K. (2016). Squeezenet: Alexnet-level accuracy with 50x fewer parameters and¡ 0.5 mb model size. arXiv preprint arXiv:1602.07360.

[8] Cao, Qiong, et al. "Vggface2: A dataset for recognising faces across pose and age." Automatic Face & Gesture Recognition (FG 2018), 2018 13th IEEE International Conference on. IEEE, 2018.

[9] Joseph P. Robinson, Ming Shao, Yue Wu, Hongfu Liu, Timothy Gillis, Yun Fu Visual Kinship Recognition of Families in the Wild IEEE Transactions on pattern analysis and machine intelligence (TPAMI) Special Issue: Computational Face, 2018

[10] Joseph P. Robinson, Ming Shao, Handong Zhao, Yue Wu, Timothy Gillis, Yun Fu Recognizing Families In the Wild (RFIW): Data Challenge Workshop in Conjunction with ACM MM 2017 Proceedings of the 2017 Workshop on Recognizing Families In the Wild, 2017

[11] Shuyang Wang, Joseph P. Robinson, Yun Fu Kinship Verification on Families In The Wild with Marginalized Denoising Metric Learning 12th IEEE Conference on Automatic Face and Gesture Recognition, 2017

[12] Joseph P. Robinson, Ming Shao, Yue Wu, Yun Fu Families in the Wild (FIW): Large-scale Kinship Image Database and Benchmarks ACM on Multimedia Conference, 2016

[13] Xia, Siyu, Ming Shao, Jiebo Luo, and Yun Fu. "Understanding kin relationships in a photo." IEEE Transactions on Multimedia 14, no. 4 (2012): 1046-1056.

[14] Xia, Siyu, Ming Shao, and Yun Fu. "Kinship verification through transfer learning." In Twenty-Second International Joint Conference on Artificial Intelligence. 2011.