KINSHIP DETECTION USING DEEP NEURAL NETWORK BASED ON RESNET50

Group No. 29

Tulika Manoj Awalgaonkar - 56485612
Asrar Zaman - 89965659
Abhishek Jha - 29968380
Karan Rao - 45601884

ABSTRACT

Through this research, we present a comparison study between different architectures that we have used for the purpose of Kinship Detection/Verification. Kinship verification has become an increasingly interesting and challenging problem in the field of AI, specifically the field of Computer Vision, and has numerous real-world applications such as forensic genealogy, cases involving missing people, and social media analysis. To identify whether two people are related, we have used three different neural network architectures trained on the Faces in the Wild dataset by Northeastern SMILE Lab. These architectures include - a transfer learning model which uses ResNet50 as the base, ResNet50 with improved concatenation operations, and finally a combination of the previous model with multiple dense layers.

1 INTRODUCTION

Kinship verification refers to the process of determining whether two people are related or not by looking at their facial appearance. The idea behind this process is that two people who are related to each other have a higher degree of facial similarity than people who are not related. More recently, this has become an interesting problem in the field of AI and Computer Vision which finds real-world applications in crime-solving, forensics, social media analysis, and such.

An ample amount of research and effort is being put into exploring this topic because although the problem might sound simple, it is far from it. There are several challenges posed by the problem of kinship verification. Extraction of facial features from images of faces using computers and determining whether a relationship exists between a pair of people by comparison of these features in itself is quite a daunting task. Further, there also exist problems related to variations in attributes of such images such as lighting, background, proximity, and facial expressions. Another common problem is the one of identifying a married couple since, without any contextual information, it is extremely difficult to know if two people are married or not simply because they do not share any DNA which makes their facial features radically different from one another.[1]

Through the research conducted for this project, we have performed a comparative analysis of three different neural network architectures to solve the problem of kinship verification. The models are as follows:

- VGGFace - ResNet50
- VGGFace - ResNet50 + improved concatenate operations
- VGGFace - ResNet50 + improved concatenate operations + multiple dense layers

2 RELATED WORK

2.1 Convolutional Neural Network

CNN is a neural network that has the capability to detect patterns or features and makes sense of them. That is the reason why CNN is useful for image analysis. It has hidden layers called convolutional layers. Convolutional layers receive input, perform mathematical computation on the input, and output the transformed input to the next hidden layer or output layer. Pattern detection in initial layers is basic such as finding corners, edges, etc. but as hidden layers progress, the pattern becomes more complex like a dog, cat, etc.

2.2 Transfer Learning

It is the use of information used to solve one problem and use the solution or calculated to solve another problem. [2] We know that in Deep Learning, initial layers are used to detect basic patterns and as the transformed input progresses through each layer, patterns become more complex or whole. That is why transfer learning can use the data gathered from previous layers as per need. For example, early features from neural networks trained to detect dogs can be used to train a model for detecting X-Rays.

2.3 Siamese Neural Network

Getting more data is not always the solution for a better prediction. For solving problems like facial recognition and signature verification, we need an architecture/model like that of a Siamese network which can get us better predictions with few numbers of images. A Siamese Neural Network is a type of neural network architecture of two or more subnetworks that accept distinct inputs which are joined by a common function at the top.[3] These subnetworks have the same weights and parameters (basically, the same configuration). A Siamese Network compares its feature vectors to find the similarity of inputs.

2.4 Binary cross-entropy loss

Binary cross-entropy, also known as log loss, is a function that compares the predicted probabilities to the actual class output which is binary i.e., it can be either 0 or 1. After that, the score is calculated based on the distance from the expected value (how close or far from the actual value).

$$H_p(q) = -\frac{1}{N}\sum_{i=1}^{N} y_i \cdot log(p(y_i)) + (1 - y_i) \cdot log(1 - p(y_i))$$

Figure 1: Binary Cross-Entropy / Log Loss

In the above function, y denotes the label, and p(y) is the probability of the label. This function returns a value, which evaluates the prediction. If the value is high, then it is a bad prediction, otherwise, when the value is low, the prediction is good.

2.5 Adam Optimizer

Adam Optimization algorithm is an optimization algorithm used to update network weights iteratively based on training data and is used instead of classical stochastic gradient descent. It is a computationally efficient algorithm that requires little memory and is appropriate for problems that are large in terms of parameters and/or data and non-stationary objectives.

3 Dataset definition

We have used the Northeastern SMILE Lab - Recognizing Faces in the Wild dataset[4] from Kaggle. Since we are comparing the above three models, we have fixed the validation dataset.

- train.zip - It is the training set and has been divided into multiple folders which represent Families (Fxxxx) and each folder has a sub-folder of individuals (MIDx).
- train_relationships.csv - It contains pairs of related family members.
- test.zip - It is the test set that contains the unknown images of unknown individuals from an unknown family.
- sample_submission.csv - It contains the final result which shows whether those pairs of images are from the same family or not.

4 Models

For the given Kinship detection problem, we came up with 3 different deep learning-based approaches. Different approaches used are as follows:

4.1 Transfer learning using ResNet-50 architecture

For the first model, we used the transfer learning approach on the VGGFace2 model developed by researchers at the Visual Geometry Group at Oxford [5]. Transfer learning is a popular approach in deep learning where parameters of the model trained on a large dataset can be reused to solve a related problem. The extracted weights and features of the state-of-the-art pre-trained models are used for the new task to improve its performance.[6] For this project, we are using transfer learning with a pre-trained network (in this case ResNet-50) to encode each (224x224x3) face image into a 4096-dimensional feature vector. Since ResNet has more than 134 million parameters (until the 4096-encoding layer), it may not be wise to train and fine-tune the entire network, so for simplicity, we freeze layers of the Vggface model. Next, a few custom fully connected layers are added on top of it. Finally, we use the sigmoid activation function to calculate the probability of classification. Binary cross-entropy is used as the loss function with Adam being the Optimizer. The model trained for 50 epochs and the weights of the model with best validation accuracy and the weights of the model at the 50th epoch are saved. The accuracy of the final model is about 76

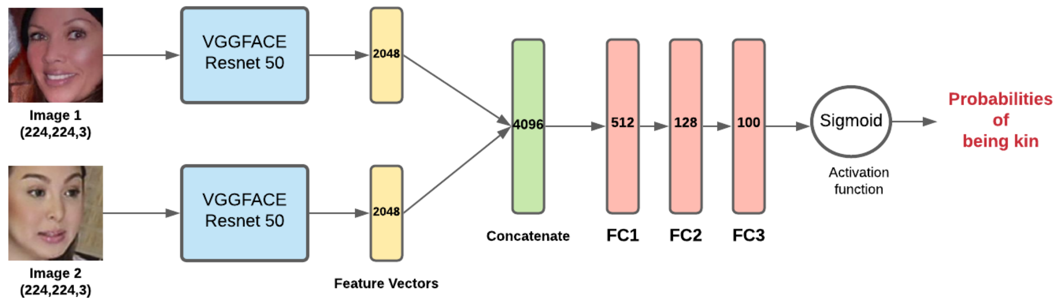

Figure 2: ResNet 50

4.2 ResNet-50 with improved concatenation operations

In this approach, we use a deep siamese network to classify images as Kin and Non-kin. Firstly, we generate feature vectors of dimensions 4096x1 from vggface ResNet-50 from pairs of face images whose dimensions are scaled down to 224x224x3. We then pass these extracted feature vectors to multiple layers which perform some mathematical operations for concatenating them such as multiplication, difference of squares, and the square of the difference between the two feature vectors. Finally, the output from these functions is passed on to a fully connected layer after which we use a sigmoid activation function(similar to the previous model) to calculate the probabilities of whether the pair of people in the input images are related to each other or not. Similar to the previous model, we use Binary cross-entropy as the loss function with Adam as the Optimizer. This model is also trained for 50 epochs and the weights with the best validation accuracy are saved (50th epoch). The accuracy of the final model is about 85

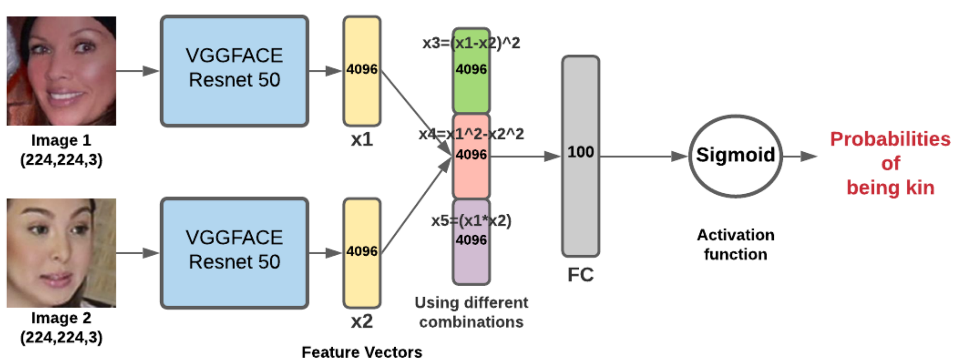

Figure 3: ResNet50 with improved concatenation

4.3 ResNet-50 with improved concatenation operations and multiple dense layers

Similar to the previous model, we perform image preprocessing and generate feature vectors using the scaled pairs of images, We again perform the concatenation operations to generate a single feature vector. Then, the fused features are fed into a fully-connected network and a sigmoid activation function to obtain the similarity score between two faces, which is used to verify the kinship [7]. The major difference between models 2 and 3 is that model 2 uses a single fully connected layer whereas model 3 uses multiple fully connected layers before feeding the output to the sigmoid function. Further, this model has been trained for 100 epochs and the weights with the best validation accuracy are saved. The added advantages of having multiple fully connected layers are:

- Dimensionality reduction.
- Improvement in model accuracy (85%-87%).

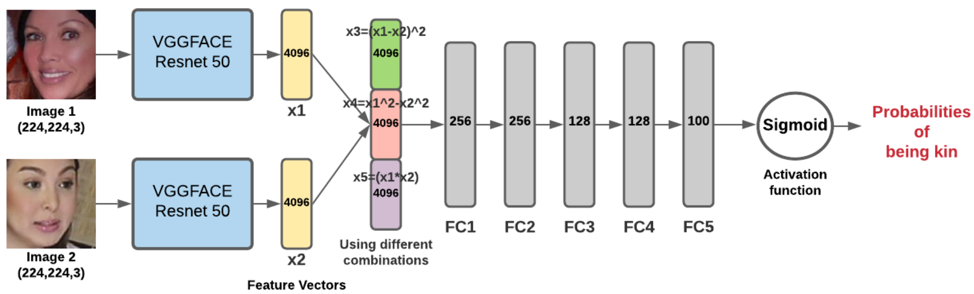

Figure 4: ResNet50 with improved concatenation and multiple dense layers

5 Visualizations

This section is divided into two subsections: the first shows a sample image and its prediction, and the second section shows graphs of training and validation accuracy along with the loss. Figure 5 shows two sample images from the validation dataset; We can also see the accuracy and the prediction of those two images. Our models give a value of >0.5 when the images are related.

5.1 Sample Image Classification

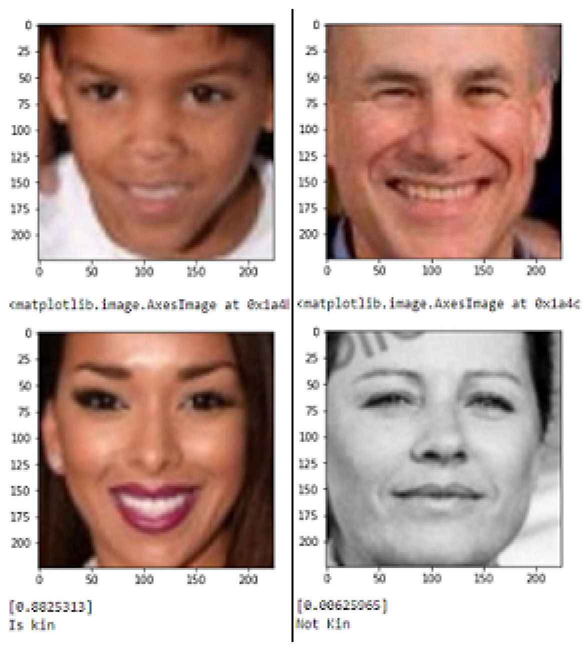

Figure 5: Sample kin classification

5.2 Accuracy and Loss graphs

5.2.1 ResNet50

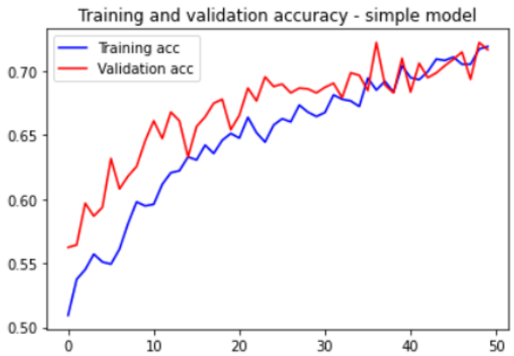

Figure 6: Accuracy: ResNet50

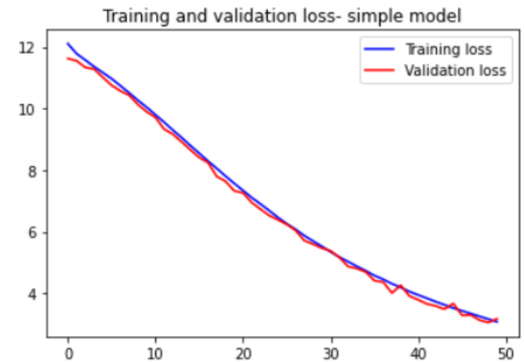

Figure 7: Loss: ResNet50

5.2.2 ResNet50 with improved concatenation

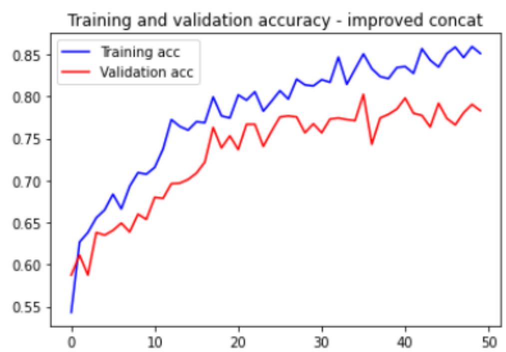

Figure 8: Accuracy: ResNet50 with improved concatenate

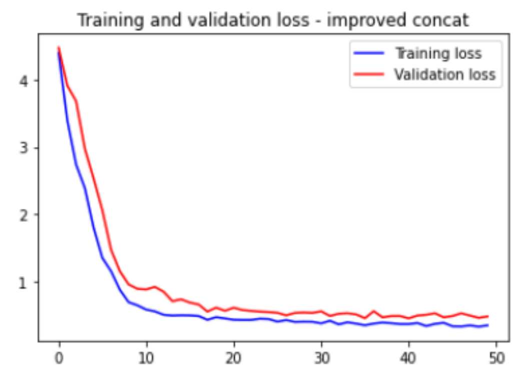

Figure 9: Loss: ResNet50 with improved concatenate

5.2.3 ResNet50 with improved concatenation and multiple dense layers

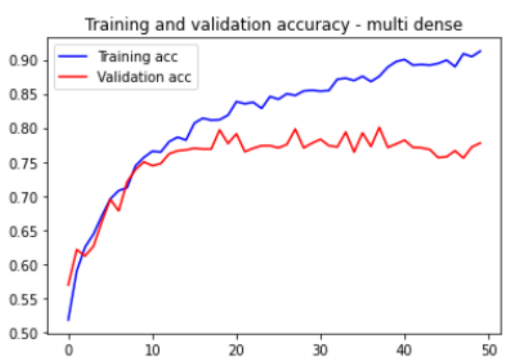

Figure 10: Accuracy: ResNet50 with improved concatenate and multiple dense layers

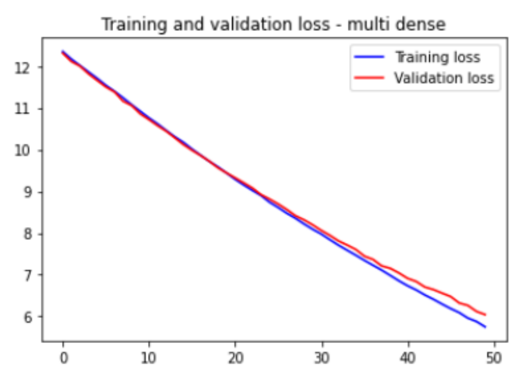

Figure 11: Loss: ResNet50 with improved concatenate and multiple dense layers

6 Results

This section includes the Accuracy and Loss tables for each model followed by the score on Kaggle. The prediction accuracy is obtained by populating the sample_submission.csv provided by Northeastern SMILE Lab's competition and submitting it on Kaggle. As seen in the table below, our third model with improved concatenations and multiple dense layers give us the best accuracy.

6.1 Accuracy results

Table 1: Accuracy (%) obtained from the models

	Training Accuracy	Validation accuracy	Prediction accuracy
ResNet50	71.94%	72.25%	76.5%
ResNet50 - improved concatenation	85.12%	80.25%	85.2%
ResNet50 - multi dense	94.13%	80.06%	87%

6.2 Loss results

Table 2: Loss obtained from the models

	Training Loss	Validation Loss
ResNet50	3.0747	3.1676
ResNet50 - improved concatenation	0.3464	0.4787
ResNet50 - multi dense	4.6796	5.1171

6.3 Kaggle Score

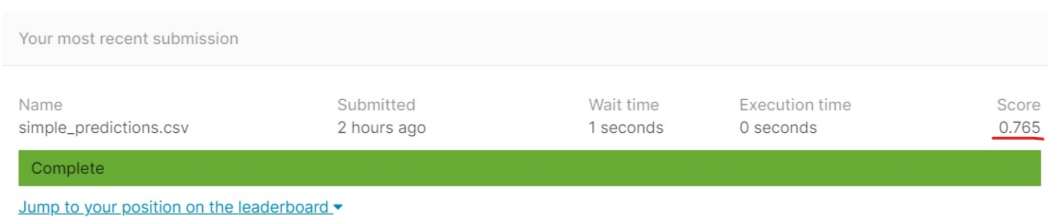

Figure 12: kaggle score of ResNet50

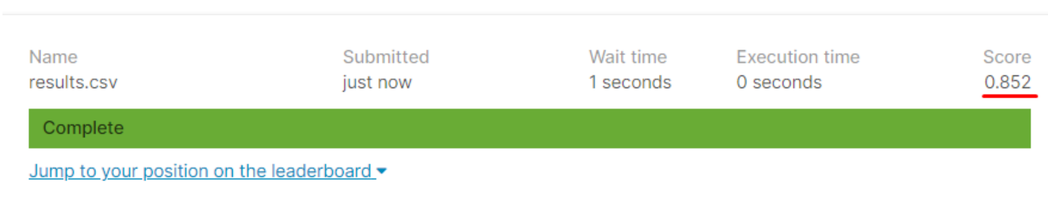

Figure 13: kaggle score of ResNet50 - improved concatenation

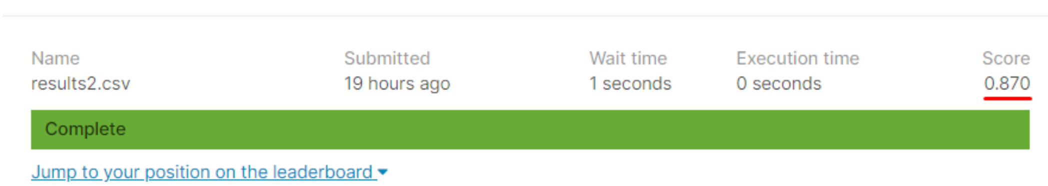

Figure 14: kaggle score of ResNet50 - multi dense

7 Conclusion

In this project, we have created three neural network models that perform kinship detection given two faces. We have used VGGFace based ResNet50 architecture and improved accuracy by introducing Global Max Pooling and Global Avg Pooling concatenates operations and by experimenting with the number of dense layers.

From this research, we have found that introducing multiple dense layers leads to overall better accuracy. This is understandable because the dense layers keep selecting better features for kinship detection. We also found that changing the way we concatenate our ResNet50 feature vectors leads to better accuracy in the testing phase.

References

[1] Mohammed Almuashi, Siti Mohd Hashim, Dzulkifli Mohamad, Mohammed Alkawaz, and Aida Ali. Automated kinship verification and identification through human facial images: a survey. *Multimedia Tools and Applications*, 76, 01 2017.

[2] Chuanqi Tan, Fuchun Sun, Tao Kong, Wenchang Zhang, Chao Yang, and Chunfang Liu. A survey on deep transfer learning, 2018.

[3] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning, 2020.

[4] Joseph P. Robinson, Ming Shao, Yue Wu, Hongfu Liu, Timothy Gillis, and Yun Fu. Visual kinship recognition of families in the wild. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(11):2624–2637, 2018.

[5] Siyu Xia, Ming Shao, and Yun Fu. Kinship verification through transfer learning. IJCAI'11, page 2539–2544. AAAI Press, 2011.

[6] Kafeng Wang, Xitong Gao, Yiren Zhao, Xingjian Li, Dejing Dou, and Cheng-Zhong Xu. Pay attention to features, transfer learn faster cnns. In *ICLR*, 2020.

[7] Jun Yu, Mengyan Li, Xinlong Hao, and Guochen Xie. Deep fusion siamese network for automatic kinship verification. *CoRR*, abs/2006.00143, 2020.