Siamese Neural Network for Face Recognition

Lifeng Shi, Guanyu Lai
Purdue University
Course: CS 578 - Statistical Machine Learning

***Abstract*— Face recognition and verification has long been a major topic that researchers have been working on in the last decade. While it is fairly simple for humans to identify the difference between people regardless of pose, lighting, and expressions, it can be difficult for computers to bypass these noise factors and grasp important facial features to make correct classifications.[11] For this project, we want to build a machine learning model that is able to capture facial features and is able to output 'same' or 'different' by looking at two pictures of two same/different individuals. More specifically, we incorporate a Siamese Convolutional Neural Network that takes a pair of images and compute their similarity score as evaluation metric. Once the network is tuned, it is able to take unseen pairs of images and determine whether they represent identical human faces.**

This report will hence be separated in the following sections: a literature review of Siamese network structure and related work, methodologies, experiments with Labelled Faces in the Wild(will be referred as LFW in the entire report), and Conclusion.

I. Literature Review

A. Siamese Network

Siamese Network structure was first implemented by Bromley et al. in 1994 to verify images of signatures [1]. It has then been widely used in image and text verification. A Siamese neural network consists of two identical sub-networks which takes a pair of distinct inputs and outputs two vectors as feature representation for each input. Then, the network computes a distance function of the two output. If the distance is within a certain threshold, then the two inputs are classified 'same', otherwise they are classified 'different'[9]. Since Siamese sub-networks are identical, the number of parameters and weights are shared, which guarantees that the two similar images should be mapped to similar locations in feature space by their respective sub-networks. [9] As a result, the Siamese structure is perfect for performing image verification tasks.

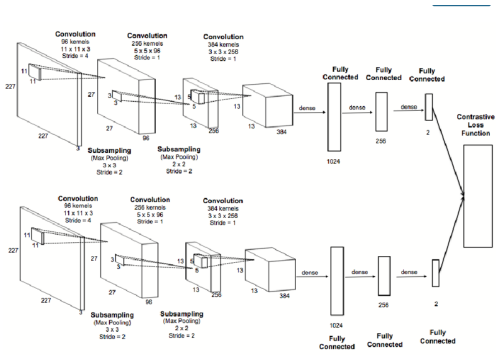

Fig. 1. An example of the Siamese Architecture

In 2005, Chopra and Lecun applied a Convoluted Siamese Neural Network to face verification tasks using the AT&T database of faces [3]. They used a contrastive loss function that combines the partial loss function of a genuine pair with the partial loss function of an imposter pair, and took average over the entire training set [3]. Their approach achieved impressively low false accept rate as well as false reject rate [3]. Besides that, Bukovcikova et al. confirmed the effectiveness of a Siamese architecture for face verification when they adopted the AlexNet architecture for each subnet and achieved 85% success rate [2]. We include an example of Siamese network architecture in Figure 1[14].

Other than that, Taigman et al. also explored Deep Siamese neural networks in the process of inventing Deep Face [18]. They used a trainable scaled absolute difference as distance measure, and used standard cross-entropy loss to train the network [18]. Their work achieved 96.17% when trained on Social Face Classification dataset and validating on the LFW dataset [18]. This is already

comparable to human performance and is one of the most efficient algorithms for face recognition today.

B. Face Recognition and Verification

In 1970s, Goldstein, Harmon, and Lesk [19]used 21 specific subjective markers including lip thickness and hair color in order to identify faces automatically. But the actual biometrics were computed manually. In 1980s to 2000s, mathematical tools were widely used in facial recognition. PCA, which applies linear algebra to make a low-dimensional representation of biometric features is used[8].But it was not until 1991 that the automatic facial recognition was applied. But the progress is still low due to pure techniques and low computational ability.

What makes facial recognition popular is the rise of CNN, convolutional neural network. CNN is composed of multiple convolutional layers and ReLUs with pooling and other features [12][15]. The convolutional layers are viewed to obtain the spatial feature of a given data, thus it is most successful in image recognition. In 2012, Hinton and his student Alex won the first prize in ImageNet Large Scale Visual Recognition Challenge because of ImageNet, one kind of CNNs[10]. Since then, facial recognition has been a heated topic and the accuracy rate went up to 99.5%. Among all the CNNs, some models are outstanding. The FaceNet got 97.8% on its testing dataset [16], VGG-Face was almost the same[13]. The Inception Resnet is viewed as one of the biggest break-through in CNN and its accuracy rate is 99.3% on testing data[17]. These networks are widely used in apps and other possible fields.

II. Methodologies

A. Convolutional Neural Network

Convolutional neural networks are more efficient for image classification compared to traditional fully connected neural networks because they have less connections and parameters, therefore resulting in less training time, but still have comparable performance.

For our project we implemented convolutional neural networks within the Siamese architecture that follows the structure suggested by Chopra et al. [3] with fine tuning. To save training time, we downsized our image data from $250 \times 250 \times 3$ to $64 \times 64 \times 3$ while cropping the images so that we only focus on the center of the face. Our model consists of four convoluted layers and followed by a fully connected layers. Our CNN structure is as follow:

1) first convoluted layer: 12 kernel filters with size 3 by 3. Add max pooling layer with pool size 2 by 2.
2) second convoluted layer: 24 kernel filters with size 5 by 5. Add max pooling layer with pool size 2 by 2.
3) third convoluted layer: 48 kernel filters with size 6 by 6.
4) fourth convoluted layer: 48 kernel filters with size 3 by 3.
5) dense layer: 60 parameters

The network outputs two vectors than we then use as input to the contrastive loss function.

We used Rectified Linear Unit (ReLU) as activation functions for all convoluted layers and the dense layer. For training, we used adam optimizer with a learning rate 0.0001.

$$f(x) = max(0, x) \tag{1}$$

To avoid overfitting, we also used dropout layers.

B. Siamese Network

1) Contrastive Loss: Let X_1, X_2 be a pair of input vectors, and let Y be the label assigned to this pair, where Y = 0 if the pair is disimilar and Y = 1 if the pair is similar. [5] Then the loss function is:

$$L(W, Y, X_1, X_2) = (1 - Y)\frac{1}{2}(D_w)^2 + (Y)\frac{1}{2}max(0, m - D_w)^2 \tag{2}$$

Here W is the learned parameter. G_w is a parametric function that maps the input vector from high dimensional space to low dimensional space. D_w is a function that calculates the Euclidean distance between X_1 and X_2 such that:

$$D_w(X_1, X_2) = \|G_w(X_1) - G_w(X_2\|_2 \tag{3}$$

And m is a margin > 0 that defines a radius around $G_w(X)$, where we have set it to be 1 because it is the most used margin [5][2].

Like all distance-based loss, contrastive loss ensures that similar pairs are embedded closer together, and dissimilar pairs are further away. [5]

C. Accuracy measure

Since the Siamese CNN outputs a similarity score for each image pair, we need to select a threshold to determine which pairs are similar and which pairs are dissimilar. In our case we selected 0.04.

For our accuracy measure we used recall (also named sensitivity):

$$Recall = \frac{TruePositive}{TruePositive + FalseNegative} \quad (4)$$

III. Experiments

A. Data Description

1) Training and Testing data: The data will be retreated from the Labeled Faces in the Wild(LFW) database, which can be found at http://vis-www.cs.umass.edu/lfw/. It is free to download and available to all that are interested. The LFW data base consists of 13233 images of 5749 people, where 1680 people have two or more images.[7] Each person is given a unique name(ID), no two person should be given the same ID in this database. Most images are colored with size of 250×250 while some images are in greyscale. [7]

The LFW aims to address the problem where given a pair of images, whether a machine can determine if these images represent the same individual, which is perfect for the purpose of our project.

The training set consists of 2200 images where (with 1100 pairs of matched images and 1100 pairs of unmatched images), and the rest of the 1000 images should be the validation set (with 500 pairs of matched images and 500 pairs of unmatched images). The images in training set and validation set are completely different, that way falsely high accuracy is avoided.

Huang et al. discover that "through image alignment, by detecting facial feature points on the image and then warping these points to a canonical configuration" [6], the face verification accuracy can greatly increase. Upon that discovery, they created the 'deep-funneled' version of the LFW data set, by using a 'combination of unsupervised alignment and unsupervised feature learning, specifically by incorporating deep learning into the congealing framework', and they have found that the deep-funneled LFW achieves 9% higher accuracy score compared to the original LFW [6]. For this project, we will be directly using an open-source repository called lfw-fuel to download the deep-funneled lfw and use this data to train our CNN model [4]. To ensure that our training data is large enough, we augmented the training pairs by flipping them along the vertical axis. Now our training data contains 2200 matched pairs and 2200 unmatched pairs.

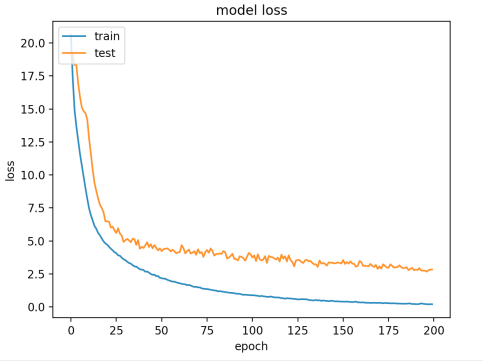

Fig. 2. Training loss plot[1]

B. Results

We input 4400 pairs of images of size $64 \times 64 \times 3$ as training pairs and 1000 pairs of images as validation pairs to our Siamese CNN model. The total number of parameters is 22,290. After 250 epochs of training, though the loss still decreases slowly, the accuracy remains around 83% (See Figure.2).

References

[1] Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Säckinger, and Roopak Shah. Signature verification using a "siamese" time delay neural network. In *Proceedings of the 6th International Conference on Neural Information Processing Systems*, NIPS'93, pages 737–744, San Francisco, CA, USA, 1993. Morgan Kaufmann Publishers Inc.

[2] Zuzana Bukovcikova, Dominik Sopiak, Milos Oravec, and Jarmila Pavlovicova. Face verification using convolutional neural networks with siamese architecture. pages 205–208, 09 2017.

[3] Sumit Chopra, Raia Hadsell, and Yann LeCun. Learning a similarity metric discriminatively, with application to face verification. In *Proc. of Computer Vision and Pattern Recognition Conference*. IEEE Press, 2005.

[1]Even though both train loss and validation loss still decrease after 200 epoch, We only include this plot that's trained for 200 epoch because accuracy stops improving.

[4] Dribnet. lfw-fuel. https://github.com/dribnet/lfw_fuel, 2018.

[5] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 2*, CVPR '06, pages 1735–1742, Washington, DC, USA, 2006. IEEE Computer Society.

[6] Gary B. Huang, Marwan Mattar, Honglak Lee, and Erik Learned-Miller. Learning to align from scratch. In *NIPS*, 2012.

[7] Gary B. Huang, Manu Ramesh, Tamara Berg, and Erik Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical Report 07-49, University of Massachusetts, Amherst, October 2007.

[8] M. Kirby and L. Sirovich. Application of the karhunen-loeve procedure for the characterization of human faces. *IEEE Trans. Pattern Anal. Mach. Intell.*, 12(1):103–108, January 1990.

[9] Gregory Koch, Richard Zemel, and Ruslan Salakhutdinov. Siamese neural networks for one-shot image recognition. 2015.

[10] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'12, pages 1097–1105, USA, 2012. Curran Associates Inc.

[11] Erik Learned-Miller, Gary B. Huang, Aruni RoyChowdhury, Haoxiang Li, and Gang Hua. *Labeled Faces in the Wild: A Survey*, pages 189–248. Springer International Publishing, Cham, 2016.

[12] Yann LeCun, Yoshua Bengio, and Geoffrey E. Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

[13] Omkar M. Parkhi, Andrea Vedaldi, and Andrew Zisserman. Recognition. In *BMVC*, pages 41.1–41.12. BMVA Press, 2015.

[14] Sanjeev Jagannatha Rao, Yufei Wang, and Garrison W. Cottrell. A deep siamese neural network learns the human-perceived similarity structure of facial expressions without explicit categories. In *CogSci*, 2016.

[15] Jürgen Schmidhuber. Deep learning. *Scholarpedia*, 10(11):32832, 2015.

[16] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. *CoRR*, abs/1503.03832, 2015.

[17] Christian Szegedy, Sergey Ioffe, and Vincent Vanhoucke. Inception-v4, inception-resnet and the impact of residual connections on learning. *CoRR*, abs/1602.07261, 2016.

[18] Yaniv Taigman, Ming Yang, and Lior Wolf. L.: Deepface: Closing the gap to human-level performance in face verification. In *In: IEEE CVPR*, 2014.

[19] Jesse Davis West. A brief history of face recognition, August 2017.