Grenze International Journal of Engineering and Technology, June Issue

Face Verification using Twin Convolutional Neural Networks

Dhruva Bhattacharya[1], Aditya Gajool[2], Sujay Banage[3], Shantanu Bhute[4] and Dr. Deipali. V. Gore[5]
[1-5]PES Modern College of Engineering, Department of Computer Engineering, Pune, India
Email: dhruva.bhattacharya@moderncoe.edu.in, {aditya.gajool, sujay.banage, shantanu.bhute, deipali.gore}@moderncoe.edu.in

***Abstract*—Nowadays various Deep learning techniques are used to make computers learn and work with representations of images. One of the important tasks in this respect is learning the similarity between two given images. This can be accomplished using Siamese Neural Network, which finds the similarity score of the given image pair. In this paper the focus is on the Siamese Convolutional Neural Network (Siamese CNN) for the purpose of face verification. The Siamese network ranks similarity between a pair of samples. It uses twin Neural networks, in this case twin Convolutional Neural Networks (for Siamese CNN). The One-Shot learning feature of the Siamese CNN makes it highly efficient against many machine learning and deep learning methods.**

***Index Terms*— Computer vision, Siamese Neural network, Siamese CNN, Convolutional Neural Network, One- Shot learning.**

I. Introduction

Machine learning is used to achieve state-of-art-performance in various applications. However, these algorithms tend to break down when forced to make predictions about data for which very little supervised information is available. One particularly interesting task is classification under the restriction that we may only observe a single example of each possible class before making a prediction about a test instance. This is called one-shot learning. [3]

A. One-Shot Learning

One-shot learning or few shot learning is the most important feature of the Siamese neural Networks. Most machine learning-based object categorization algorithms require training on hundreds or thousands of examples, one-shot leaning aims to classify objects from one, or a few **positive** samples. If the standard machine learning method is trained with only a few **positive** samples, it will suffer from severe overfitting. In comparison, our method is based on Siamese network which build the architecture with twin convolutional neural networks and share the same parameters which helps avoiding overfitting.
Here positive samples mean data in the positive dataset.

B. Similarity Learning

In this project the principle of similarity learning has been used. Similarity learning is an area of supervised machine learning where the goal is to learn a similarity function that finds the similarity between two image by

Grenze ID: 01.GIJET.9.2.353

measuring how similar or related two objects are using a similarity value / score.

C. Siamese Convolutional Neural Network

The term "Siamese" evolved from the concept of conjoined twins that are physically connected to each other. [4] Hence the Siamese NN is also called as twin Neural Network. Siamese NNs make use of two identical sub-networks sharing the same features and parameters. The sub-networks can be either Multilayer Perceptrons (MLPs), Convolutional Neural Nets (CNNs), etc.

In this paper the Siamese Convolutional Neural Network has been implemented. Model explanation (2.2.2).

II. RELATED WORKS

- The Siamese neural networks were applied to conduct one-shot classification on the Omniglot dataset to identify whether two input images were in the same category [3]. The paper shows an accuracy of 92.0% once optimizing the said Siamese neural networks. In addition, it demonstrated the learned features at one-shot learning capable to discriminate between different characters and aggregate for the same category. [1]
- Siamese CNN has been used on the Tatt-C dataset for tattoo recognition, i.e to know whether the sample is a tattoo or not. Here, both the contrastive and triplet loss functions are used, with the triplet having an upper hand as regards to performance [4].
- Siamese CNN has also been used for multi-pedestrian tracking systems. To match a pair of detected pedestrians. [4]
- DeepFace system is a multi-stage approach for face recognition proposed in [6]. One of the experiments trained the Siamese network using standard cross entropy loss and backpropagation of the error. It predicted the similarity by L1-distance between two extracted features and determines whether two faces belong to the same person. The accuracy of the DeepFace-Siamese networks reaches about 96.17%. [6]

III. MOTIVATION

The aim of this project is to introduce the application of a very powerful deep learning architecture – The Siamese Neural Network for face verification. The one-shot learning feature makes the Siamese architecture very special as it requires less data samples of the images of the faces to be verified by the system. In this paper the Siamese CNN architecture has been used for Face verification.

IV. PROBLEM DOMAIN

A. Deep Learning

Deep learning is a subset of Machine learning which uses Deep Artificial Neural Networks. These Neural Networks have multiple hidden layers apart from the input and output layer. These extra layers help in refining the neural networks and increases the accuracy.

The main purpose of Deep learning is to make machines simulate the behavior of the human brain. It basically teaches computers to do what comes naturally to humans i.e to learn by examples (data). Deep learning drives many AI based applications and services and is the key technology behind driverless cars, virtual assistants, chatbots, etc.

Deep learning models enhance automation and can even achieve state-of-the-art accuracy which sometimes even surpasses human-level performance.

B. Computer Vision

Computer vision is a field of AI where systems are trained to derive meaningful information from digital images and videos and take necessary actions or make predictions based on this data. It includes methods for acquiring, processing, analyzing and understanding digital images and extraction of high – dimensional data from the real world. It basically tries to enable computers to replicate the human visual system.

V. PROBLEM DEFINITION

Researching the Siamese Neural Nets, comparing its accuracy with other Machine Learning models and implementing a face verification system on python using the Siamese CNN model which is a type of Siamese Neural network.

VI. STATEMENT

- **Input**: Given a live image, find out if the image is similar to the ones present in the dataset by calculating the similarity scores between the pairs of images.
- **Output:** Verify the images based on the similarity score. If the similarity score is above the threshold, then the output is 1 or verified, else it is 0 or unverified.

VII. INNOVATIVE CONTENT

The One-Shot learning Feature of the Siamese Network makes it highly efficient compared to other Machine learning and Deep Learning methods (like CNN, Knn, Random Guessing etc.).

In this project 3 datasets have been used to train and test the model. This helps achieving a higher accuracy than other ML methods. Here the Anchor, Positive and Negative Datasets are used. The Anchor and Positive datasets belong to the same class during the model building stage. After model is built , the final project takes user image and compares it to the Anchor dataset. More information on Datasets (A.1,A.3)

VIII. PROBLEM FORMULATION

A. Design Methodology

Dataset Preprocessing

In this system three datasets are used for training the model, the positive dataset, the negative dataset and the anchor dataset.

- The anchor and the positive datasets contain the same class of data. These datasets contain the images of faces to be verified as 1 or True by the model.
- The negative dataset [9] is a large dataset which contains images of random people, which is used to tell the model who not to verify. The negative dataset used here contains over 13000 images of faces collected from the web. Each face is labelled. Image dimension = 250x 250.

Before being fed to the model each image from all the datasets is Normalized and resized for better results.

1. In normalization the value of each pixel of the image is reduced to a range between 0 to 1. The pixel values range from 0 to 256. So, by dividing all the values by 255 will convert it to range from 0 to 1.
2. The images are rescaled into 105 x 105 pixels as the size of images may vary and this also improves the focus on the faces of people.
3. The images in the positive dataset could also be resampled for more accuracy.

Building The Model

The Siamese CNN network contains twin CNN networks which share the same weights and parameters.

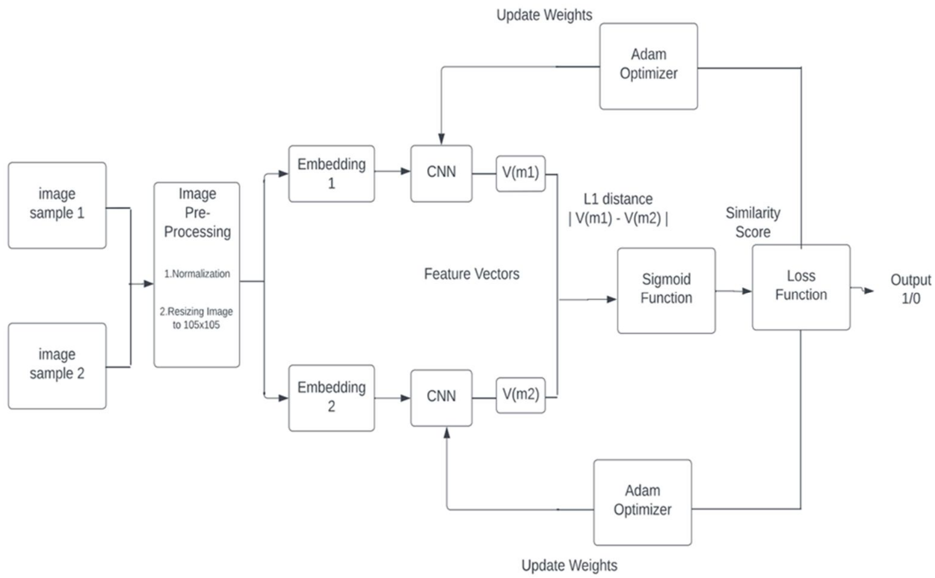

Figure 1 Model Overview

Figure 1 is the model overview and also the dataflow diagram for the Face verification system. This is the final trained and tested model where the image sample 1 is the user image from the camera and the image sample 2 are images from the anchor dataset.

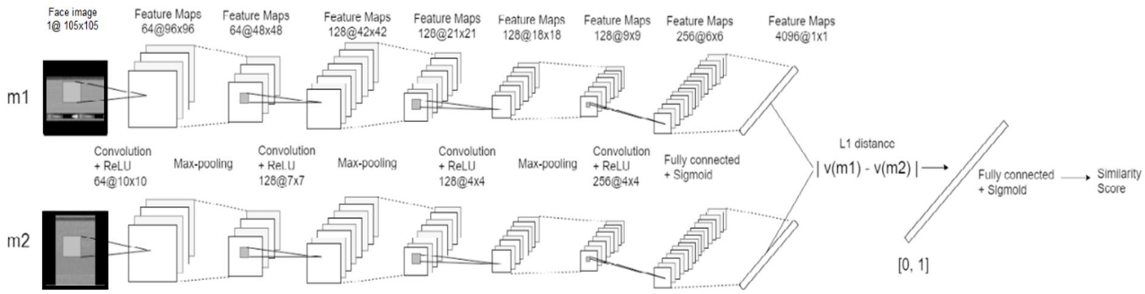

Figure 2 The Structure of Siamese Convolutional neural networks [1]

The Convolutional neural Network (CNN) is mainly composed of the convolution layer, pooling layer and the fully connected layer (Fig 2) [5]. This paper adopts Siamese convolutional neural networks to implement twin CNNs which share the same weights and parameters (fig 1) [1]. The sequence of convolutional layers in twin CNNs use filters of different size and fixed stride to extract the feature maps. The output of each convolution in the first three layers is fed into the Rectified Linear Unit (ReLU) activation function and max-pooling operation with stride two. The final convolutional layer is followed by a fully connected layer with sigmoid activation function, where the feature maps are flattened into a single vector. In contrast to typical CNNs used in standard classification, one additional customized layer is added to calculate the Manhattan distance (L1 distance) between two extracted feature vectors. And then the result is passed through a fully connected layer with the sigmoid function and units set to one. The final output signifies the similarity score of the image pair in the range of [0, 1]. [1]

Training The Model

The first step is to divide the datasets into batches for training and testing the model. The model is then trained on these mini batches of images. Then, the networks use adaptive moment estimation optimizer (Adam) [9] to update the weights during each mini batch of training iteration to increase the accuracy of the model. The anchor dataset is compared to the negative and positive datasets in order to train the model.

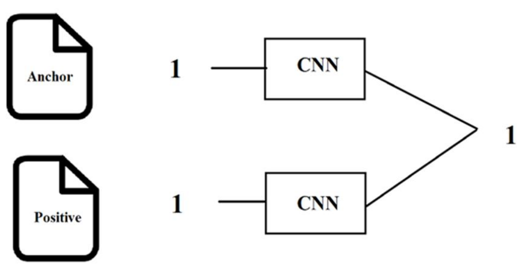

Figure 3 Training model to verify positive data

Fig 3 shows how The Siamese CNN is trained using the Positive and the anchor datasets (3.1.1). This basically tells the model which images (faces) to verify (what's correct) using the positive dataset.

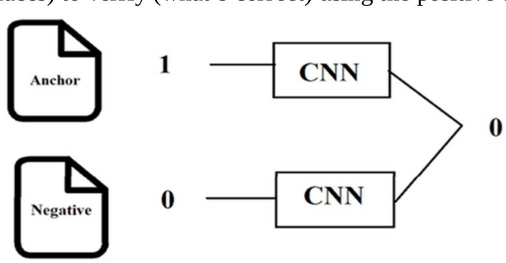

Figure 4 Training model against negative data

Fig 4 demonstrates how The Siamese CNN is trained using the Negative and the anchor datasets (3.1.1). This basically tells the model which images (faces) not to verify (what's incorrect) using the negative dataset.

Testing The Model

After the model is trained and built it is tested on parts of data belonging to the same dataset it has been trained on. The data which is not used in the training stage is used to validate the Siamese CNN. One of the advantages of testing on new dataset is to prove that the trained model has no overfitting problems.

Various performance metrics can be used to test the model , such as accuracy score, Precision, recall ,etc. The Sklearn library on python provides the above mentioned functions to train the performance of the model.

B. Mathematical Model

Reference [3]

Loss Function: Here M represents the minibatch size, where i indexes the i th minibatch. Let $y(x_1^{(i)}, x_2^{(i)})$ be a length-M vector which contains the labels for the minibatch, where we assume $y(x_1^{(i)}, x_2^{(i)}) = 1$ whenever x_1 and x_2 are from the same character class and $y(x_1^{(i)}, x_2^{(i)}) = 0$ otherwise. We impose a regularized cross-entropy objective on our binary classifier of the following form:

$$L(x_1^{(i)}, x_2^{(i)}) = y(x_1^{(i)}, x_2^{(i)}) \log p(x_1^{(i)}, x_2^{(i)}) + (1 - y(x_1^{(i)}, x_2^{(i)})) \log (1 - p(x_1^{(i)}, x_2^{(i)})) + \lambda^T |w|^2$$

Optimization: This objective is combined with standard backpropagation algorithm, where the gradient is additive across the twin networks due to the tied weights. We fix a minibatch size of 128 with learning rate η_j , momentum μ_j , and L_2 regularization weights λ_j defined layer-wise, so that our update rule at epoch T is as follows:

$$w_{kj}^{(T)}(x_1^{(i)}, x_2^{(i)}) = w_{kj}^{(T)} + \Delta w_{kj}^{(T)}(x_1^{(i)}, x_2^{(i)}) + 2\lambda_j |w_{kj}|$$
$$\Delta w_{kj}^{(T)}(x_1^{(i)}, x_2^{(i)}) = -\eta_j \nabla w_{kj}^{(T)} + \mu_j \Delta w_{kj}^{(T-1)}$$

where ∇w_{kj} is the partial derivative with respect to the weight between the j^{th} neuron in some layer and the kth neuron in the successive layer.

IX. Problem Solving

A. Project Planning:

- Topic selection
- Generating the Problem Statement.
- Understanding the inputs and outputs of the project.
- Assigning tasks to teammates.

B. Defining Requirements

- Defining the software, Hardware, functional and Data requirements for the implementation of the project.
- Installing required moules : TensorFlow, Matplotlib, numpy , Pandas ,Keras and OpenCV.

C. Fulfilling Data Requirements

- The project requires 3 datasets, the positive, negative and anchor dataset.
- The negative data set was obtained from an online source [9].
- The positive and anchor dataset are to be made by the developers.
- Preprocessing all the datasets by performing Normalization i.e dividing each pixel by 255 and getting them in a range between 0-1.
- Resize images from negative dataset to 105 x 105.
- Split the datasets into training ,testing and validation sets.

D. Model Building

- Refed to research papers for Architecture diagrams and make necessary changes (if required). Made class, use case and state diagram for better understanding of the project.
- Get the required libraries (tensorflow, keras, numpy,sklearn).
- Build the SNN model based on the architecture diagram (Figure 1).

- Train the model using the training dataset.
- The network was trained for a maximum of 50 epochs.

E. Testing the Model

- Use test data to test the model.
- Use various performance metrics like Precision, recall, accuracy score to check the model's accuracy.
- Make necessary changes if required.
- Test the model on live images. Face Verification Using Siamese Neural Network.

F. Deployment

- Tweak the model to get desired results.
- Link the trained and tested model to the front end.
- Deploy the model after getting it approved from the project guide.

X. RESULTS AND SENSITIVITY ANALYSIS

After training the model on images of **1 person** in the positive dataset and over 1680 face images in the negative dataset. The model was then evaluated using various performance metrics like Precision, recall and accuracy Score. TensorFlow.keras.metrics was used for evaluating the model. The model threshold was set to 0.7 i.e. similarity score above 0.7 = image verified.

TABLE I. EVALUATION RESULTS

Performance Metric	**Score**
Precision	0.98
Recall	1.0
Accuracy Score	1.0

On evaluating the model against live input, the model gave 100% accuracy for true negatives i.e. The outputs which were expected to be False were verified false by the system. However, in case of True Positives the Model gave an accuracy of 77.5%. Images were taken in multiple backgrounds.

Some positive inputs were being wrongly classified by the model. On further evaluation it was observed that the background noise and lighting greatly affected the results. Under appropriate lighting the model gave almost perfect accuracy. On re testing the model under proper lighting conditions the model predicted the True Positives with an accuracy of 90%.

With a constant background and appropriate lighting, the accuracy of the model would even reach over 97%.

XI. DATA MODEL

The model was tested against live input in various lighting conditions and different backgrounds. Some of the instances of live testing are given below in table II.

While live testing the threshold 0.7 gave the best outputs. So the results are for the model with 0.7 as threshold.

Pls Note the similarity score is in the range of 0.0 to 1.0.

Input details:

- Image size = 105 x 105.
- Captured using 1080p webcam.

Output details:

- Similarity score of image to images in database.
- Verification output : true / false.

XII. COMPARISON OF RESULTS

A. Comparison of Siamese CNN to Convolutional Neural Network

The two models were compared to each other for image based recognition on the FLAVIA dataset which contains images of leaves of various plant species [10]. Various experiments were performed and the results of them are shown in table III:

TABLE II. INPUTS AND THEIR CORRESPONDING OUTPUTS

input	Expected result	Output	Reason
	True	Similarity Score: 0.91 True	
	False	Similarity Score: 0.12 False	Image not present in the database.
	True	Similarity Score: 0.72 True	

Experiment 1: Accuracy Achieved for different dataset sizes.

TABLE III: ACCURACY ACHIEVED FOR DIFFERENT DATASET SIZES [10]

Experiment	Dataset Size			Siamese CNN		CNN	
	Training	Validation	Testing	Top -1	Top - 3	Top - 1	Top - 3
E_5	4	1	10	68.7	85.6	34.4	61.6
E_{10}	8	2	10	67.5	88.1	34.7	60.0
E_{15}	12	3	10	64.1	89.1	36.6	63.8
E_{20}	16	4	10	67.2	93.1	64.1	88.7
E_{25}	20	5	10	66.9	92.2	71.6	91.6
E_{30}	24	6	10	72.8	92.8	77.8	93.7

From Table III we can see that for 20 or less images per species, the top-1 and top-3 accuracy of the Siamese CNN is consistently better than the CNN.
Experiment 2: Accuracy/species for 20 Costa Rican tree species.
The top-1 and top-3 accuracy were obtained for both Siamese CNN and CNN for the 20 new species of plants.

TABLE IV: ACCURACY/SPECIES FOR 20 COSTA RICAN TREE SPECIES [10]

Species	Siamese CNN		CNN	
	Top – 1	Top – 3	Top – 1	Top – 3
Average accuracy	56	81	48	70

The table-IV summarizes the top-1 and top-3 accuracy obtained by Siamese CNN and CNN for 20 new species. Higher accuracies were shown by the Siamese CNN. The average overall accuracy of Siamese CNN was much higher than the CNN model , the top-1 and top -3 accuracy of Siamese CNN were 8% and 11% higher respectively.

B. Comparison with KNN and Random Guessing

The Performance of KNN, Random guessing and Siamese CNN was evaluated using the N- Way One shot learning tasks. Where it was found out that for every case the Siamese CNN gave better accuracy than the KNN and Random Guessing methods. [1]
In Table III,
SCNN (TR) = Siamese CNN (Training Dataset)
SCNN(TE) = Siamese CNN (Testing Dataset)
KNN = K- Nearest Neighbor
RG = Random Guessing

TABLE V: THE ACCURACY OF N-WAY ONE-SHOT LEARNING FOR DIFFERENT METHODS [1]

Model	N-way														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
SCNN (TR)	100.0	95.3	93.3	87.3	82.3	68.6	74.6	70.6	68.6	59.3	62.0	62.6	60.6	55.3	60.0
SCNN (TE)	100.0	92.6	84.0	72.0	75.3	76.6	70.0	60.6	62.6	60.6	54.0	47.3	52.6	42.6	42.6
KNN	100.0	66.6	60.0	41.3	40.0	44.6	38.0	27.3	25.3	24.6	24.6	20.6	22.0	15.3	24.6
RG	100.0	50.0	33.3	25.0	20.0	16.6	14.2	12.5	11.1	10.0	9.1	8.3	7.7	7.1	6.7

XIII. JUSTIFICATION OF THE RESULTS

When the Siamese neural network was used for one shot classification on the omniglot dataset to identify, whether two input images were in the same category. The model accuracy went upto **92.0%** after optimization.[1].

Our Siamese CNN model achieved an accuracy of 91.25% at 0.7 threshold while live testing. Which is extremely close to the desired output.

XIV. CONCLUSION

Researched and studied a very powerful Neural Network – the Siamese Neural Network, and the applications of Siamese CNN in the field of computer vision. Also compared the model to other Machine learning models for image verification/ recognition and noted that the model gives 22% higher accuracy than the KNN method ,40-44% higher accuracy than Random Guessing and 8 – 11% higher accuracy than the Single CNN method. Based on the research conducted a Face Verification system was trained and built using the Siamese CNN model which uses the concept Similarity learning to verify images. The model takes a live face image as an input and extracts it into a vector and then checks the similarity between the two images. And based on the similarity scores attained the model verifies the image.

FUTURE WORK

Many improvements can be made to reduce the background noise from the user inputs, this would make the model extremely accurate. The verification system faced problems due to the inconsistent lighting. Connecting a lighting source and automating it to keep the lighting consistent would solve the issue. The future research work of this project includes training the model using different loss functions, activation functions, etc. and comparing their results.

REFERENCES

[1] 2019 ,Malware Image Classification Using One-Shot Learning with Siamese Networks , by: Shou-Ching Hsiaoa , Da-Yu Kaob, Zi-Yuan Liua , Raylin Tsoa, ELSEVIER

[2] 2022, Face Recognition Method Based on Siamese Networks, IEEE Access ,Auth :CUNLI SONG AND SHOUYONG JI College of Software, Dalian Jiaotong University, Dalian, Liaoning, China

[3] 2016, One-Shot classification, Siamese Neural Networks for One-Shot Image Recognition, by Gregory Koch, Department of Computer Science University of Toronto

[4] 2020, Published in IEEE Xplore .A Survey on Applications of Siamese Neural Networks in Computer Vision, INCET , Auth : Abhilash Nandy, Sushovan Haldar, Subhashis Banerjee and Sushmita Mitra, Indian Institute of Technology, Kharagpur,India

[5] 2021,Zewen Li, Fan Liu , Wenjie Yang,Shouhen Peng and Jun Zhou. ", A Survey of Convolutional Neural Networks: Analysis, Apps & Prospects" ,IEEE Transactions

[6] DeepFace: Closing the Gap to Human-Level Performance in Face Verification, in: 2014 IEEE Conference on Computer Vision and Pattern Recognition, Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, 2014

[7] 2016, Siamese Networks Features for Image Matching, IEEE , By: Iaroslav Melekhov, Juho Kannala and Esa Rahtu

[8] Intuition of Adam Optimizer:https://www.geeksforgeeks.org/intuition-of-adam-optimizer/

[9] Dataset : Labelled Faces in the wilds http://vis-www.cs.umass.edu/lfw/

[10] Using a Convolutional Siamese Network for Image-Based Plant Species Identification with Small Datasets , MDPI open Access journals, by : Geovanni Figueroa-Mata, and Erick Mata-Montero

[11] Open Cv website: https://opencv.org/