Covenant Journal of Informatics & Communication Technology. Vol. 7 No. 2, Dec. 2019
ISSN: p2354 – 3566 e 2354 – 3507 DOI:
An Open Access Journal Available Online

Signature Verification Using Siamese Convolutional Neural Networks

Chika Yinka-Banjo & Chinonso Okoli

University of Cape Town, South Africa, UNILAG
okolichinonsojohnbosco@gmail.com, cyinkabanjo@unilag.edu.ng

Received- August 11, 2019 Accepted – November 21, 2019
Date of Publication: December, 2019

Abstract—This research entails the processes undergone in building a Siamese Neural Network for Signature Verification. This Neural Network which uses two similar base neural networks as its underlying architecture was built, trained and evaluated in this project. The base networks were made up of two similar convolutional neural networks sharing the same weights during training. The architecture commonly known as the Siamese network helped reduce the amount of training data needed for its implementation and thus increased the model's efficiency by 13%. The convolutional network was made up of three convolutional layers, three pooling layers and one fully connected layer onto which the final results were passed to the contrastive loss function for comparison. A threshold function determined if the signatures were forged or not. An accuracy of 78% initially achieved led to the tweaking and improvement of the model to achieve a better prediction accuracy of 93%.

Keywords/Index Terms—Biometric, Brute Force, Empirical, Metric, CNN, SCNN.

1. Introduction

A lot of cyber-crime attacks are due to poorly employed methods in identity verification. Current approaches to biometric verification are either very expensive to build and maintain or are inaccurate and efficient. Signature verification on the other hand, when compared to other biometric forms of identification is considerably cheaper and easier to maintain (Yang, Z., Oathes, D.J., et al. 2018.). However, this method has not been fully embraced as it is deemed archaic and unnecessary.

In today's world where a lot of computational power is needed to use biometric identity techniques like facial and fingerprint recognition due to the amount of training datasets involved in training such systems (Kulikajevas, A., Maskeliūnas, R., et al, 2019), the Siamese neural network becomes quite handy when cost is put into consideration.

Computer Vision as a method for image classification can be applied to signature verification as these tasks are quite similar (Smith, C., McGuire, B., et al, 2006). There are ways to process and extract information from given signatory images. One might use processes like Support Vector Machines or Deep Learning Architectures like Convolutional Neural Networks (Mnih et al. 2015).

However, these methods have their respective bottlenecks. Support Vector Machines have a slight manual feel to it. The programmer or software architect would have to manually watch out for discriminant features and extract or note them (H., Luger, 2009). As for current existing deep learning architectures, tasks like these would require a lot of training data and thus, a lot of computational power. In both cases, there are downsides that can be avoided since classification is the main aim. Hence, the reason Siamese Convolutional Neural Networks are used for this task.

This architecture which employs two base sister neural networks and a distance based metric function to compute similarity level between inputs can be used in classifying these inputs. This architectural technique is employed in the development of the network in this project because it yields similar results for far less power and resources, which is an important feature in any architectural model.

This project was faced with a specific problem of old signature verification methods employed in some organizations and firms. These methods apart from being old are mostly inaccurate and hard to implement. An example of such is the manual cross-checking of a signature against previous signatures stored in a folder by a person. Since people are non-reliable and wear out, a person can accidentally misjudge a signatory.

In the end, automated techniques which do not require so much resources to execute are a great idea towards solving the issues of biometric verification.

2. Literature Review

2.1 Machine Learning

A computer program is said to learn from experience E with respect to some task T and some performance measure P, if its performance on T, as measured by P, improves with experience E. (Mitchell, 2008), (Russell, P. Norvig. 2009).

Machine Learning (ML) is the application of learning and self-correction mechanisms in machines to enable them solve problems rationally. Machine Learning is mostly a study and improvement of the different algorithms that can facilitate this learning process in artificial Intelligence. Machine Learning can be supervised, unsupervised and semi-supervised. (Nadkarni et al., 2011)

2.1.1 Turing Test

Alan Turing sought to answer the question "Can Machines think?"

(Turing, 1950), (Davalo, E., Naïm, P., et al, 1993) with an experiment. The Turing test was developed in 1950 to test a machine's ability to exhibit intelligent behaviour close to that of humans. He proposed a gaming scenario involving three players, a computer and two humans. These three players were isolated from one another, and one of them, an interrogator had the task of figuring out which one of the other two players was a computer. The interrogator had to ask questions to both these players in the attempt to achieve this goal. If the interrogator, at the end of this experiment is unable to distinguish between these players, then the computer is said to be intelligent.

Alan Turing predicted that by the year 2000, machines and computers would carry out this assignment (the turing test) without being detected. However, this is not the case today. No computer or machine has come close to achieving this feat.

2.1.2 Supervised Learning

Supervised Machine Learning is a type of learning in which a model is provided with labels, meaning or extra information as to what kind of data is given to it (Attia, Z.I. et al, 2019). In supervised learning, every decision the model makes is based on the information attributed to the percepts it has recognized. No extra learning is done outside this confinement (Mitchell, 2008).

2.1.3 Semi-Supervised Learning

In Semi-supervised Learning, the model is provided with a large set of unlabeled data (data without meaning or information attached to it) and a small set of labeled data. This is to permit a non-bias nature when the models deal with prediction (Hao, Y., Colak, R., 2015). Semi-supervised Learning is the most widely used kind of learning because it satisfies both areas. If instances are given with known labels (the corresponding correct outputs) then the learning is called supervised. (Kotsiantis, 2007).

2.1.4 Unsupervised Learning

Unsupervised Learning is a kind of learning in which no labels nor extra information are provided along with the dataset. Here, the models extract features based on information they had gotten from varying patterns in this data (Waydo, S. and Koch, C. 2008). This kind of learning is most suitable for Artificial General Intelligence (AGI). Unsupervised Learning is usually applied in instances where we want to discover unknown classes of items from a given sample. (Jain et al, 1999).

2.1.5 Linear Regression

Regression is a method of narrowing a particular target value based on independent predictors, it is a forecasting technique which is used to figure the relationship between dependent and independent variables. Regression analysis is a technique used in statistics for investigating and modeling the relationship between variables (Douglas Montgomery, Peck, & Vinning, 2012).

Linear Regression is a kind of regression analysis in which there is a linear relationship between the independent variable and the dependent variable. This relationship allows one to forecast values if given an independent value (H. Winston, M. Hill, 1975).

2.1.6 Logistic Regression

Logistic Regression is used when the dependent variables are categorical and the independent variables are continuous (Nadkarni, P.M., Ohno-Machado, L. et al, 2011). An example is the classifying of a card transaction as fraudulent or not in which case it is binary, you can only have a yes or no answer. Logistic Regression uses a Sigmoid function to map any real value between 0 and 1. After mapping, a decision boundary (threshold) is used as a pass mark for what is a yes or a no (Schneider et al., 2018).

2.2 Deep Learning

Deep Learning is a class of machine learning that takes the approach of modeling the architecture of human reasoning, neural networks (Mitchell, 2008). A deep learning architecture consists of different neurons firing impulses/data to other neurons in the same network (Green, C.S., Kattner, F., et al, 2015), (Hu, X. and Balasubramaniam, P. 2008). Deep Learning is the latest approach to artificial intelligence as it provides modern solutions to problems (B., Jahne, H. Haussecker, 1999). It has been applied for solving various problems by reserchers()

2.2.1 Artificial Neural Networks (ANN)

Artificial Neural Networks is the underlying architecture of the deep learning model. Neural Networks make use of neurons, biases, weights, activation functions, layers and a lot of other methodologies and concepts to model the way the human brain works, (Lillicrap et al., 2016). These networks or models learn to perform tasks simply by looking through examples without being programmed for specific cases (Mahanta, J. 2017).

2.2.2 Convolutional Neural Networks (CNN)

Neural network is successfully applied in various domains (Alade et al. 2017, Kulikajevas, et al., 2019, Ogwueleka, et al. 2014, Ogwueleka, et al. 2015, Okewu et al. 2018, Abayomi-Alli et al. 2019, Alhassan & Misra, 2011). Convolutional Neural Networks are a class of neural networks that specialize in processing matrices of data. Examples of such data are visual data such as images and videos among others. A CNN or ConvNet architecture typically has three type layers: a convolutional layer, a pooling layer and a fully connected layer. The first precise and accurate CNN was in the year 2010 (Krizhevsky, Hinton, 2010), (LeCun, Y., Bengio, Y. and Hinton, G. 2015) which won the ImageNet classification competition that year.

The Convolutional Layer carries out the main task of this network. It performs a dot product between two sets of matrices or grids. One of the sets being a learnable parameter called the kernel (Silver, D., Hubert, T., Schrittwieser, J., et al. 2018). The other set is a portion of the input or receptive field. The result of this operation is a feature map which contains certain characteristics of the main input. This feature map is then passed through a Rectified Linear Unit (ReLU) which is an activation function that maps all negative real numbers to zero.

The pooling layer simply reduces the spatial size of the output of the convolutional layer to avoid heavy computation. It does this by deriving a statistical summary of the feature map either by Max-pooling or Average-pooling (Rashid, T., 2010). Max-pooling replaces a grid area by the largest pixel value in the grid while Average-pooling takes a mean representation of the grid area (Kriegeskorte, N. 2015).

The fully connected layer is a set of layers containing neurons connected to every other neuron in the preceding and succeeding layers. It is computed just like the Fully Connected Neural Networks using a matrix multiplication followed by the addition of a bias (Fraser, C. 2003).

2.2.3 Siamese Neural Networks (SNN)

A Siamese Neural Network is a class of Artificial Neural Networks whose architecture is made up of two identical sub neural networks sharing weights and working in tandem (Yoshida, S. 2011). Identical in the sense that they share the same configuration with same parameters and parameter updating is the same and done in parallel. This kind of network is especially good for tasks that involve finding similarity or the relationship between two or more sets of data. The identical networks in this architecture work as they normally would outside this architecture (Cao, Y., Jiang, T. and Girke, T. 2010.). However, their outputs are compared over a metric distance function or a loss function such as the triplet loss function or the contrastive loss function, (Banino et al. 2018).

2.3 Contrastive Loss

Contrastive Loss is a distance based function as opposed to the prediction-error based loss functions used in the modern neural network architectures (Banino, A., Barry, C., et al 2018), (Zhou, Z. and Schwenker, F. 2013). Contrastive Loss groups inputs based on the similarity of their semantics. It is a pair-wise function, meaning that it works with pairs of inputs unlike the triplet and centre loss functions which use triplets and point-wise calculations respectively (Fitzgibbon, A., Taylor, C., et al, 2006). Hadsell used the contrastive loss function to learn the parameters of a function through pulling neighbours together and pushing non-neighbours apart (Hadsell et al, 2006).

2.4 Existing Related Technologies

2.4.1 Imagenet Classification With Deep Convolutional Neural Networks (2012)

In 2012, a paper showed for the first time how Deep Convolutional Neural Networks can outperform other methods at Image Classification (Krizhevsky, Sutskever and Hinton, 2017). This paper was released by Alex Krizhevsky, a computer science student of Toronto and his PhD advisor Geoffrey Hinton along with IIya Sutskever after they won the ImageNet Large Scale Visual Recognition Challenge in 2010. They built a CNN, which they called AlexNet with just eight (8) layers. The AlexNet had its first five (5) layers to be convolutional and pooling layers which were then followed by fully connected layers.

The AlexNet became the breakthrough for Image classification as well as opened doors for other Computer Vision tasks in the world of Artificial Intelligence (Michalski, R. 2014).

However, in 2014, Geoffrey Hinton gave a talk on the short-comings on Convolutional Neural Networks. He said it was unfortunate that CNNs work so well because they have flaws that he believes won't be easy to get rid of (Brewka, G. ECAI 2006). He mentioned that back propagation was an inefficient way of learning because it requires a lot of data. In his talk, he proposed a new way of learning, pattern recognition from capsule representation. He also wrote a code on MatLab (wasn't optimized until later versions) that could compete with the CNN of today.

Geoffrey Hinton was one of the first to cite the problems with CNNs although he had no ready solution to these problems.

2.4.2 Learning A similarity Metric Discriminatively, With Application To Face Verification

Summit and Chopra wrote a paper on Siamese Neural Networks stating primarily scenarios where the number of categories or features to identify are very large and not known during training (Sumit, Chopra et al, 2017) . These Mathematical Sciences students from New York University (NYU) tested this network on the Purdue/AR face database which is known for variation amongst its data and the network performed well. The measure of performance relying mostly on speed and reliability.

Furthermore, probabilistic models weren't used in this research, so instead of assigning a normal probability to the variations being modeled, they used Energy-Based Models (EBM) which assigns normalized energy instead. They judged the performance of their system on the classes; the percentage of false rejected to false accepted. Triplet Loss wasn't used in this network, Contrastive Loss was used instead because they wanted to reduce the chances of the network assigning constant values to some input images instead (Graham, I. 2003). Convolutional Neural Networks used as the independent networks were also used to map plain images to points in a low dimensional space to fish out a similarity metric.

2.4.3 Similarity Learning with (or without) Convolutional Neural Networks.

In their work, they also talked about metrics as a form of learning. In this case, comparisons which dictate similarity or dissimilarity are made between inputs. The inputs go through a series of operations and transformations to enable faster computation and better accuracy before their outputs are passed through a similarity function like the Euclidean distance function (Hinton, G. and Sejnowski, T. 1999). The result of this is a real number denoting how close or far off the inputs are from each other. Depending on the magnitude of this number, is the similarity value. A large number means the inputs are not similar, far off apart. A small number on the other hand means the inputs are similar to each other (M. Chatterjee, 2010).

This paper was a contribution to the on-going research on Siamese Neural Networks and its applications. Siamese Neural Networks at the time was a field of interested research and showed promises regarding solutions to the short-comings of CNNs; their need of large training data-sets. Their research was based on a similar research by Sumit Chopra, Raia Hadsell and Yann LeCun.

Moitreya Chatterjee went on to further do research on Discriminative Descriptors for Local Patches. A method which learns discriminative representation of image data and patches from different 3D view points. This is a very interesting area of research as it not only improves precision of prediction, but also gives room for distinct solutions. Of course, a problem with this area of study is the required computational power of its implementation.

2.4.4 On the Generalized Distance in Statistics

This paper was based on a project comparing distance metric systems of sorts, which was later used as a determinant function in a Fingerprint Recognition task. Although this particular project was more theoretically inclined than practical, it however proves that the idea of using metrics as a comparison function between two entities dated back to the 90s. This is the whole backbone of the Siamese networks of today, (Aurelien., 2017), (P. Baldi,1993).

Professor Pierre Baldi, a Professor from the University of California, Irvine who used this project in his Fingerprint Recognition Network used the fact that a P-variate normal population can be represented by a density cluster in P-dimensional space, and can always be transformed to a set of independent variates. P. C. Mahalanobis in his paper in 1936 mentioned that they may differ in their mean values or position of the clusters. However, two such dissimilar normal populations can be superposed by a function, a squeeze and a rotation. In layman terms, two non-identical populations likely have a translational connect.

This proved useful in many classification algorithms as far as comparison between and among classes are concerned. This is so because no two classes are exactly the same and so therefore, it is always a plus to be able to derive a relationship between these classes in order to establish a threshold.

2.4.5 Deepface: Closing the Gap to Human-Level Performance in Face Verification.

In 2014, the Institute of Electrical and Electronic Engineers (IEEE) held the 27th conference on Computer Vision and Pattern Recognition (CVPR) at Columbus, Ohio. In this conference, DeepFace was talked about and shed light on.

DeepFace is facial recognition network that uses a nine-layered deep learning architecture in identifying human faces in images (IEEE, 2014). It has an outstanding 97.25% accuracy, which is just 0.28 percent less than a human being. Compared to all other previous face detection programs and/or software, including the 85% accurate FBI’s Next Generation Identification system, the DeepFace is superior at facial recognition and identification.

The DeepFace uses a siamese network architecture in its design and implementation. In their paper, they stated that this deep network has more than 120 million parameters with several locally connected layers without weight sharing. This is different from the normal Siamese Convolutional Neural Network architecture because the Deepface simply uses metrics rather than Convolutional layers.

According to the MIT Technology review, DeepFace is a ‘significant advancement’ because of the deep learning technique. Facebook’s scientists and researchers trained DeepFace from the company’s image data, four (4) million facial images from 4,000 distinct people.

What makes DeepFace so accurate is the different scenarios where people tweak a few parameters to improve learning in Artificial Intelligent systems. Facebook didn't hold back in that area. The four stages of Deep Learning, especially Computer Vision and Pattern Recognition are:

I. Detect,
II. Align,
III. Represent, and
IV. Classify.

DeepFace kicked up the align and represent stages by a notch. For the training of DeepFace, a three-dimensional (3D) model of the average human being was used. The system re-positions the face to its front view and then maps this reorientation using modeled neural networks. If DeepFace comes with a similar mapping from two distinct images beating a particular threshold, it validates it as the same face.

This as you recognize, is the principle of a Siamese Neural Network architecture. Its accuracy of detection is based on the degree of similarity or dissimilarity of a set of images.

2.4.6 Siamese Neural Networks for One-Shot Image Recognition

Gregory Koch cited a scenario where an agent or a model is restricted to observing only a single example of each of the classes before used against their test instances. He stated, for obvious reasons that the conventional Neural Networks would fail at various test cases because they didn't have enough training data to learn or work with (G. Koch, 2008).

The model used in this project was trained and tested on a subset of the omniglot data-set. The model was first optimized to master the verification task, which they then used to discriminate learned features.

Gregory after his experiment showed the results of the one-shot trial and not only inferred, but showed that the Siamese Neural Networks achieved an outstanding feat towards the learning based on the learning restriction (learning based on one example of each class) compared to its counterpart the Convolutional Neural Networks.

His model was a ten (10) layered twin convolutional set, including the fully connected layers, which happened to be on each set. It had three (3) convolutional layers and three (3) pooling layers which used the Max-Pooling method. These two sets of layers join immediately after the fully connected layer has done its computation. The distance metric, L1 is then computed using a metric function (Kůrková, V., Manolopoulos, Y., et al, 2017).

2.4.7 Review of Existing Algorithms for Face Detection and Recognition

Ismail and Sabri, (2009), started stating the advantages of physiological Biometric ID like facial recognition, fingerprint detection and retina scan over behavioural Biometric ID methods like keystrokes, voice recognition and signature verification. It is agreed that each passing day, physiological biometric are improved upon and people tend to focus less on behavioural metrics. I also think behavioural metrics can be very efficient if improved upon.

In his paper, he enumerated five (5) different algorithms for facial recognition; Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA), Skin Colour-based Algorithm, Wavelet-based Algorithm and Artificial Neural Networks based Algorithm. Out of these, the most similar or closest relation to this project is the Artificial Neural Networks based Algorithm because of the use of Fully Connected Layers here.

He compared these algorithms not to state which was better, but to enumerate their strengths and weaknesses (Maalej, A., Amor, B., 2011). Every research ever done on algorithms in Artificial Intelligence had at maximum, two to three algorithms in comparison.

2.5 Discussion and Comparison

Using Siamese Convolutional Neural Networks as an architecture has its advantages. It is cheaper, easier to implement and maintain and also doesn't demand as much training datatsets as other neural architectural models. This is especially a good thing for one shot tasks like facial and signature recognition since we want to be able to change as little as possible as requirements change.

For instance, imagine a firm whose security measures entail its employees signing in with face ID. When an employee leaves or a new employee is hired, requirements change and thus, the whole network has to be trained again on new set of data. This is time consuming and expensive. The Siamese architecture allows for one image sample of this new employee to be stored in the database and then a similarity check is done everytime this new employee signs in.

Compared to existing technologies, the Siamese Convolutional Network is very efficient. However, this architecture has limited abilities as they can only be used for comparison and classification tasks. This is where all other deep learning architectures in general have the upper hand.

3. Methodology

3.1 Datasets

The dataset comprised of five (5) handwritten signatures of thirty (30) people each, and in total, a number of 150 signatures. Each person amongst this group of 30 people tried forging signatures given to them by the others in the group. These forges were kept and is being used as a test measure. These signatures during training were compared with one another in random pairs to get a relationship eventually used as a threshold.

Person	Actual Signature	Forged Signature(1)	Forged Signature(2
001	00100001	00100030	00100027
002	00200002	00200026	00200025
003	00300003	00300019	00300022
004	00400004	00300018	00400017
005	00500005	00500013	00500014
006	00600006	00600008	00600007
007	00700007	00700006	00700005
008	00800008	00800002	00800001
009	00900009	00900004	00900005

Figure 1. Data sampling of images containing the first ten (10) signatories over a sheet.

Each person involved in the data gathering process were labeled accordingly from 001-030. The column “actual signature” contains the labels of the images containing the actual signatures of each person in each row. The “forged signature” columns contain all the four (4) images of forged signatures by the persons in each row. Thus, each person has at least four (4) forged signatures of his/her actual signature.

3.2 System Description

3.2.1 The Convolutional Network

The twin Convolutional Neural Networks each take in an image containing a signature as input and takes these images through different layers that carry out several computation processes. These layers are explained in details below.

3.2.1.1 The Convolutional Layers

The convolutional layers contain filters/kernels that need to be learned. Each one of these filters is convolved with the input to get an activation map containing neurons. These filters are then slid across the input to get a feature map and also update these filters/kernels. At each spatial position, a new feature is learned and the convolving process is done through a process called convolution. Convolution is the dot product between the input image and the filter/kernel to give a feature map. Mathematically, it is defined as:

$$C[m,n] = \sum_{u}\sum_{v} A[m+u,n+v] \bullet B[u,v] \qquad (1)$$

Where:

A and B are the input image and the filters respectively,

C is the feature map gotten from the convolution process, and

u and v are both unit pixel vectors.

There are two convolutional layers present in the network, each one outputting a feature/activation map. These feature maps are passed onto the RELU activation layer which zeroes all negative values and makes all positive values linear. This yields faster computation as there is less work for the model in abstraction.

3.2.1.2 The Pooling Layers

This layer reduces the size of the spatial information gotten from the feature maps. It uses MaxPooling to pick an element in a stride as a representation. MaxPooling is a type of pooling in which the element in a stride with the maximum value is chosen to represent its stride. (Rashid, 2008).

3.2.1.3 The Fully Connected Layer

A flattened vector is created in this layer which undergoes processes used in the multi-layered perceptron. Neurons are created based on the units of representation in the result of the pooling layer (O'Reilly, Zahed, 2010) and go through activation functions to yield well represented vectors.

3.2.1.4 Parameter Sharing

Parameter sharing is the sharing of weights and biases of a neuron to other neurons in a neural network (Pokharna, 2016). The whole system undergoes parameter sharing from the moment the input images are first being passed to the kernels. This is so that the sister convolutional networks do not have to generate first time weights if one has already been generated by one of the two networks.

3.2.2 The Siamese Network

The Siamese Network consists of two identical Convolutional Neural Networks with shared weights. The Convolutional Networks are connected by an energy function, Contrastive Loss which uses a distance metric as a map for similarity. This function is what determines how similar or dissimilar the images are.

3.2.3 Contrastive Loss:

The contrastive loss function is a distance-based loss function that works with pairwise inputs, determining their degree of dissimilarity (Kumar, Quora, 2018). It is a greedy function because it concentrates on pairs of measurements while discarding former or later pairs or inputs. The contrastive loss function for a single pair is given mathematically by:

$$yd^2 + (1-y)\max(margin - d, 0)^2 \quad (2)$$

Where y is the label for the inputs,
margin is the threshold used to judge the inputs,
d is the Euclidean distance between two image feature maps and is defined by:

$$d = \sqrt{(f_1 - f_2)^2} \quad (3)$$

4. Implementation and Results

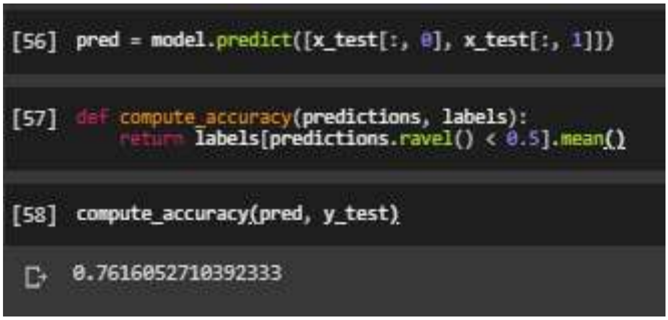
4.1 Implementation

Training datasets were gotten from Kaggle, after which pre-processing was carried out. The implementation started with the importation of the zipped processed dataset files into the Google colab environment. The needed various libraries were also imported. The next stage in the process was resizing the images to avoid mass computation.

The base convolutional networks are then created with their individual layers and their fully connected layers. After their creation, training is begun and the contrastive loss function uses the euclidean distance to calculate the dissimilarity degree.

4.2 Results

Test cases were applied after training and validation of the model. These test cases are the test portion of the dataset after it was split into training, testing and validation parts. The test phases were first carried over 13 iterations (epochs) and achieved the following results.



```
[56] pred = model.predict([x_test[:, 0], x_test[:, 1]])

[57] def compute_accuracy(predictions, labels):
         return labels[predictions.ravel() < 0.5].mean()

[58] compute_accuracy(pred, y_test)

     0.7616052710392333
```

Figure 2. Model accuracy in percentage over 13 iterations.

In line 56, the model made predictions based on what it had learned so far and assigned its results to a variable pred. This variable was then passed into a function compute_accuracy in line 57 to validate and compute its closeness to the actual test data. An accuracy of 76%, which is considered fair was achieved after this comparison.

Although it was ensured that this model didn’t overfit the training data, this wasn’t a desirable result. The accuracy achieved was below standard and could not be trusted given other datasets. Measures had to be taken to yield better accuracy as well as not overfit.

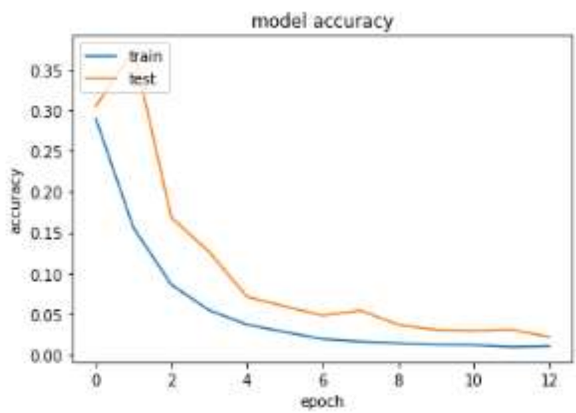


Figure 3. Accuracy to number of iterations graph of the model over 13 iterations.

During the process of the first iteration/epoch, the model is seen to move away from the training data. This is because the first weights initialized were randomized. If weights aren’t randomized, there’s a great chance the model learns the inputs as well and maps these inputs to particular repeated weights and thus, the model trains itself to memorize data. The randomization of these weights help to reduce as much bias as possible during the learning phase. As time goes on, the model learns the parameters and begins to get closer to the labels from the training data. Eventually, at epoch 12, it gets fairly close enough.

Further tweaking of the model’s parameters was made to ensure that this model doesn’t overfit, but increases its accuracy. The tweaks and steps taken are as follows:

A dropout layer was added to so as to stop the training whenever the validation error crossed a particular margin.

The learning rate was reduced by a number of 0.02 to help the validation error widen its margin.

The number of epochs(iterations) was increased to 50 to allow better error correction by the network.

The batch size was reduced to enable a slow but adequate learning.

The training process was then carried out again with these new parameters. During the training phase, the frequency of visitation of inputs with weights had increased and the model had more time to map out specific qualities pertaining to inputs. After the training phase, the validation phase ensured that testing for little amount of data gave the required actual labels. The testing phase was then carried out again to see how accurate the model had become at predicting labels. The accuracy achieved at the end of the training, validation and testing phases have the following results below.

Figure 4. Model accuracy in percentage over 50 iterations.

Again, the model computed the accuracy for the signature recognition and mapping process and came in with a 93% accuracy.

This was a desirable be gotten due to continuous training, but this is at the risk of having the model overfit due to recognizing the data. Also, the model did not overfit nor underfit as the dropout layer did not stop the training exercise due to a cross in the threshold of the validation error margin.

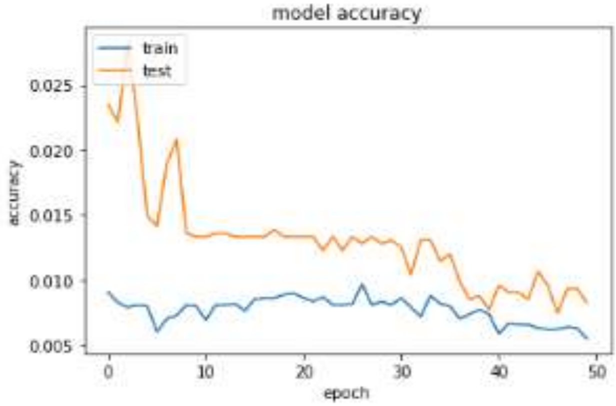


Figure 5. Accuracy to number of iterations graph of the model over 50 iterations.

In the above image, during the process of the first few iterations/epochs, the model is also seen to move away from the training data. Also, as time goes on, the model learns the parameters and begins to get closer to the labels from the training data. Eventually, at epoch 20 through to 50, it gets very close but still does not overfit.

5. Conclusion and Future Works

5.1 Conclusion

In conclusion, this project has provided a means of biometric validation of persons through the use of signatures. Using deep learning, it is not only accurate but also a trusted and tested means for secure authentication to one's data and information.

With an accuracy of 94.6% at best, this network when deployed can provide services to clients and consumers of IT products in security, since it's accuracy for average load is better than most of today's software solutions for biometric verification.

5.2 Future Works

Implementation of this network as a library to enable software developers access its use without having to write a signature verification network from scratch.

Deployment of this project along with an interface and API as a Software as A Service (SAAS) to users and clients where and when needed.

References

Attia, Z.I., Kapa, S., Lopez-Jimenez, F., et al. 2019. Screening for cardiac contractile dysfunction using an artificial intelligence-enabled electrocardiogram. Nature Medicine 25(1), pp. 70–74.

Alhassan, J. K., & Misra, S. (2011). Using a weightless neural network to forecast stock prices: A case study of Nigerian stock exchange. Scientific Research and Essays, 6(14), 2934-2940.

Abayomi-Alli, A., Abayomi-Alli, O., Vipperman, J., Odusami, M., & Misra, S. (2019, July). Multi-class Classification of Impulse and Non-impulse Sounds Using Deep Convolutional Neural Network

(DCNN). In International Conference on Computational Science and Its Applications (pp. 359-371). Springer, Cham.

Alade, O. M., Sowunmi, O. Y., Misra, S., Maskeliūnas, R., & Damaševičius, R. (2017, December). A neural network based expert system for the diagnosis of diabetes mellitus. In International Conference on Information Technology Science (pp. 14-22). Springer, Cham.

Banino, A., Barry, C., Uria, B., et al. 2018. Vector-based navigation using grid-like representations in artificial agents. Nature 557(7705), pp. 429–433.

Cao, Y., Jiang, T. and Girke, T. 2010. Accelerated similarity searching and clustering of large compound sets by geometric embedding and locality sensitive hashing. Bioinformatics 26(7), pp. 953–959.

Green, C.S., Kattner, F., Siegel, M.H., Kersten, D. and Schrater, P.R. 2015. Differences in perceptual learning transfer as a function of training task. Journal of Vision 15(10), p. 5.

Hao, Y., Colak, R., Teyra, J., et al. 2015. Semi-supervised Learning Predicts Approximately One Third of the Alternative Splicing Isoforms as Functional Proteins. Cell reports 12(2), pp. 183–189.

Ismail, N., & Sabri, M. I. M. (2009, December). Mobile to server face recognition: a proposed system for social situations in Malaysia. In Proceedings of the 8th WSEAS International Conference on Computational intelligence, man-machine systems and cybernetics (pp. 48-56). World Scientific and Engineering Academy and Society (WSEAS).

Kriegeskorte, N. 2015. Deep neural networks: A new framework for modeling biological vision and brain information processing. Annual Review of Vision Science 1, pp. 417–446.

Kulikajevas, A., Maskeliūnas, R., Damaševičius, R., & Misra, S. (2019). Reconstruction of 3D Object Shape Using Hybrid Modular Neural Network Architecture Trained on 3D Models from ShapeNetCore Dataset. Sensors, 19(7), 1553.

LeCun, Y., Bengio, Y. and Hinton, G. 2015. Deep learning. Nature 521(7553), pp. 436–444.

Mitchell, T.M., Shinkareva, S.V., Carlson, A., et al. 2008. Predicting human brain activity associated with the meanings of nouns. Science 320(5880), pp. 1191–1195.

Mnih, V., Kavukcuoglu, K., Silver, D., et al. 2015. Human-level control through deep reinforcement learning. Nature 518(7540), pp. 529–533.

Nadkarni, P.M., Ohno-Machado, L. and Chapman, W.W. 2011. Natural language processing: an introduction. Journal of the American Medical Informatics Association 18(5), pp. 544–551.

Okewu, E., Misra, S., Fernandez-Sanz, L., Ayeni, F., Mbarika, V., & Damasevicius, R. (2019). Deep Neural Networks for Curbing Climate Change-Induced Farmers-Herdsmen Clashes in a Sustainable Social Inclusion

Initiative. Problemy Ekorozwoju—Problems of Sustainable Development, 14, 143-55.

Ogwueleka, F. N., Misra, S., Colomo-Palacios, R., & Fernandez, L. (2015). Neural network and classification approach in identifying customer behavior in the banking sector: A case study of an international bank. Human factors and ergonomics in manufacturing & service industries, 25(1), 28-42.

Ogwueleka, F. N., Misra, S., Ogwueleka, T. C., & Fernandez-Sanz, L. (2014). An artificial neural network model for road accident prediction: a case study of a developing country. Acta Polytechnica Hungarica, 11(5), 177-197.

Schneider, S., Taylor, G.W., Linquist, S.S. and Kremer, S.C. 2018. Past, Present, and Future Approaches Using Computer Vision for Animal Re-Identification from Camera Trap Data. Methods in ecology and evolution / British Ecological Society 10(4), pp. 461–470.

Silver, D., Hubert, T., Schrittwieser, J., et al. 2018. A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play. Science 362(6419), pp. 1140–1144.

Waydo, S. and Koch, C. 2008. Unsupervised learning of individuals and categories from images. Neural Computation 20(5), pp. 1165–1178.

Yang, Z., Oathes, D.J., Linn, K.A., et al. 2018. Cognitive behavioral therapy is associated with enhanced cognitive control network activity in major depression and posttraumatic stress disorder. Biological psychiatry : cognitive neuroscience and neuroimaging 3(4), pp. 311–319.

Yooseph, S., Nealson, K.H., Rusch, D.B., et al. 2010. Genomic and functional adaptation in surface ocean planktonic prokaryotes. Nature 468(7320), pp. 60–66.

Russell, P. Norvig. 2009. Artificial Intelligence: A Modern Approach, Third Edition, pp. 26–77.

H., Luger, University of New Mexico, 2009. Artificial Intelligence: Structures and Strategies for Complex Problem Solving, Sixth Edition, pp. 117–190.

T. Jones. 2008. Artificial Intelligence: A Systems Approach, pp. 567–1120.

Rashid 2010. Make Your Own Neural Network, pp. 39–91.

Aurelien. O'Reilly Media, 2017 Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems 1st Edition, Kindle Edition, pp. 926–1190.

IEEE Conference on Computer Vision and Pattern Recognition, 2014. DeepFace: Closing the Gap to Human-Level Performance in Face Verification, pp. 15–34.

Smith, C., McGuire, B., Huang, T., & Yang, G. 2006, December. History of Artificial Intelligence . pp. 233–245.

Mahanta, J. 2017. Introduction to Neural Networks, Advantages and Applications, pp. 419–467.

Lillicrap, T.P., Cownden, D., Tweed, D.B. and Akerman, C.J. 2016. Random synaptic feedback weights support error backpropagation for deep learning. Nature Communications 7, p. 13276.

Mitchell, T.M., Shinkareva, S.V., Carlson, A., et al. 2008. Predicting human brain activity associated with the meanings of nouns. Science 320(5880), pp. 1191–1195.

Mnih, V., Kavukcuoglu, K., Silver, D., et al. 2015. Human-level control through deep reinforcement learning. Nature 518(7540), pp. 529–533.

H. Winston, M. Hill, 1975. The Psychology of Computer Vision. Pp. 129–150.

Jahne, H. Haussecker, P. Geissler, Academic Press, 1999. Handbook of Computer Vision and Applications, Vol. 3, Systems and Applications, pp. 945–1014.

Brewka, G. ECAI 2006. Amsterdam: IOS Press, pp.450-544.

Yoshida, S. (2011). Computer vision. Hauppauge, N.Y.: Nova Science, p.112.

Davalo, E., Naïm, P. and Rawsthorne, A. (1993). Neural networks. Basingstoke: Macmillan, p.144.

Fitzgibbon, A., Taylor, C. and LeCun, Y. (2006). Proceedings of the 2006 Conference on Computer Vision and Pattern Recognition Workshop. Los Alamitos, Calif.: IEEE Computer Society, pp.90-133.

Graham, I. (2003). Artificial intelligence. Chicago, Ill.: Heinemann Library, p.11.

Hinton, G. and Sejnowski, T. (1999). Unsupervised learning. Cambridge: The MIT Press, pp.213-240.

Michalski, R. (2014). Machine Learning. Elsevier Science, pp.908-1186.

Chopra, S., LeCun, Y., Provost, F., Bottou, L., Greengard, L. and Leahy, J. (2008). Factor graphs for relational regression. pp.34-35.

Hu, X. and Balasubramaniam, P. (2008). Recurrent neural networks. Rijek, Crotia: InTech, p.12.

Zhou, Z. and Schwenker, F. (2013). Partially Supervised Learning. Berlin, Heidelberg: Springer Berlin Heidelberg, p.999.

Fraser, C. (2003). New biometric technology takes facial 'fingerprint'. Biometric Technology Today, 11(3), p.3.

Kůrková, V., Manolopoulos, Y., Hammer, B., Iliadis, L. and Maglogiannis, I. (2017). Artificial neural networks and machine learning - ICANN 2018. p.34.

Maalej, A., Amor, B., Daoudi, M., Srivastava, A. and Berretti, S. (2011). Shape analysis of local facial patches for 3D facial expression recognition. Pattern Recognition, 44(8), pp.1581-1589.

Krizhevsky, A., Sutskever, I. and Hinton, G. (2017). ImageNet classification with deep convolutional neural networks. Communications of the ACM, 60(6), pp.84-90.

Summit, S. Chopra, X. Hou, L. Zhang (2017). Learning a similarity metric discriminatively, with application to face verification. pp. 34-37.

Chatterjee (2010). Similarity learning with (or without) convolutional neural networks. pp. 68-70.

Baldi and Y. Chauvin, (1993) Neural Networks for Fingerprint Recognition, Neural Computation,5. pp. 53-59.

IEEE Conference on Computer Vision and Pattern Recognition, 2014. pp. 1701–1708.

Nurul Huda İsmail, Mas Idayu MD. Sabri Department of System and Computer Technology (Multimedia), Faculty of Computer Science and Information Technology, University of Malaya, 50603 Lembah Pantai, Kuala Lumpur, Malaysia. Review of existing algorithms for face detection and recognition. pp. 178-179.

G., Koch, 2008. Siamese neural networks for one-shot image recognition. pp. 120-122