International Baccalaureate Diploma Programme

**Extended Essay**

Subject: Computer Science HL

THE POSSIBLE ROLE OF NEURAL NETWORKS IN VERIFYING HANDWRITTEN SIGNATURES

Research Question: To what extent can a simple image recognition neural network verify if a signature is genuine or forged?

Word Count: 3955 + 44 (doubled lines of code) = 3999

Table of Contents

[Introduction 3](#_Toc534276696)

[Design and Method 4](#_Toc534276697)

[Research Plan 5](#_Toc534276698)

[Input 6](#_Toc534276699)

[Construction and Calculation 8](#_Toc534276700)

[Normalisation by an Activation Function 9](#_Toc534276701)

[Training the Network 10](#_Toc534276702)

[The Cost Function 10](#_Toc534276703)

[Backpropagation 11](#_Toc534276704)

[Gradient Descent 12](#_Toc534276705)

[Solutions to Internal Problems 13](#_Toc534276706)

[Testing the Network and Results 13](#_Toc534276707)

[Analysis of the Results and Errors 15](#_Toc534276708)

[Conclusion 18](#_Toc534276709)

[Works Cited 19](#_Toc534276710)

[Appendix 19](#_Toc534276711)

[Results table 19](#_Toc534276712)

[Signature class file 20](#_Toc534276713)

[Network File 21](#_Toc534276714)

# Introduction

A signature is an important part of our identity, something we use in legal documents to prove that it is truly us who sign a certain document as an authorisation thereof in order to validate a transaction being closed on our behalf as a form of deliberation and informed consent. Therefore, it is no wonder that throughout the history of written documents there have been many people who have tried to impersonate another person by forging the latter’s signature for their own gain. Because we cannot check each and every signature, unless we had handwriting experts analysing every document produced, signature forgery in the financial industry is still a widespread practice.[[1]](#footnote-1) There could be countless instances of forgery elsewhere in the corporate world that have cost millions regardless of elaborate systems of authentication in place, leading to costly litigations oftentimes to no avail. Therefore, the idea should be tested whether signatures could be verified by automated methods, digitally, where neural networks could be trained by examples of legitimate signatures to detect any attempted forgeries. Neural networks, being computer programs which simulate human brains by acting as neurons connected to one another and, most importantly, learning from mistakes (in the supervised instance, which is used in this case), seemed perfect for said purpose. Since even the genuine signatures of one person often deviate to a certain degree, neural networks could be the key to noticing a pattern of writing different from all others which is invisible to the untrained eye. The question is, to what extent can a simple image recognition neural network verify an electronic signature? To answer this question, an artificial neural network should be built, one trying to discover this imagined pattern that directs the system to the genuineness of a set of signatures. After completion, the network should be tested with one set of authentic and faked signatures, and its capabilities of telling them apart should be assessed. For the purpose of the experiment, the focus will be on signatures that are written on electronic tablets, e.g., for passport security, because they have a unified format and are devoid of additional variables such as pen pressure, ink amount and other factors. The code will be based on Finn Eggers’ YouTube series, and the works cited are books by neural network experts and scientific papers produced on the application of neural networks for verifying signatures and the factors that influence them.

# Design and Method

Corresponding to the central research question, the objective of this investigation is to evaluate a neural network’s ability to recognise patterns in authentic signatures and tell forgeries apart.

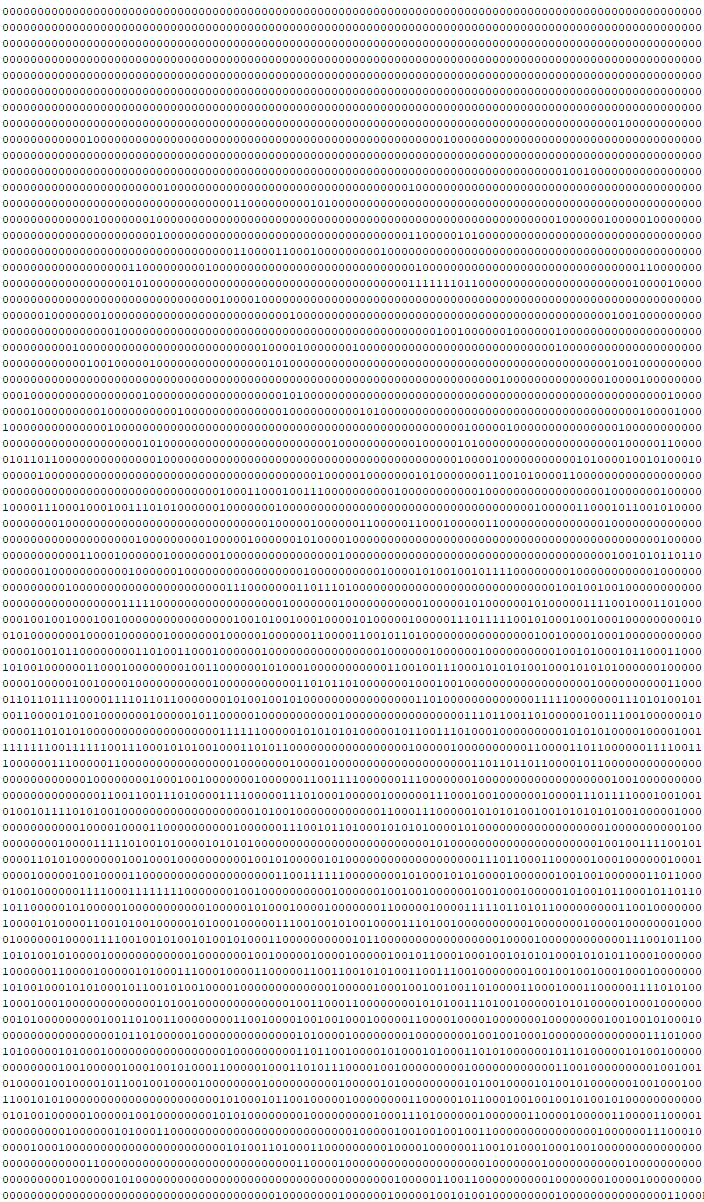
In order to answer the question, a specific neural network must be built. One which takes the input of a single image, calculates internal outputs, adjusts the calculation during the training process and, finally, gives a single output that corresponds to “1” (signifying a real signature) or “0” (signifying a forged one). This system must be able to be activated multiple times both for the training and for the testing processes. This particular deep neural network (meaning a network which has more than one hidden layer) will be built in object-oriented, class-based programming language Java, and a large part of it will be built with the code from Finn Eggers’ YouTube series “NN – Fully Connected Tutorial”[[2]](#footnote-2).

## Research Plan

The goal of this experiment is to emulate signatures written electronically, create a neural network capable of image recognition to test them and analyse the results according to the research question asked. The input images of the signatures have to be made beforehand for the network to analyse them. The network’s complexity cannot be chosen to be of a too large degree because of the limited timeframe of the experiment; therefore, the neural network has to be of a simple enough design so that it could be created, tailor-made specifically for the images it would analyse and having not too many complex parts for further maintenance of it. The simplicity of the network will, however, have an impact on the hypothesis for the answer to the research question, which is that the network would be able to tell forgeries apart from the original signature but with a rather high margin of error because the program will not notice the small details.

Once the network is trained, the next phase in the plan would be the testing stage. Two signatures would be input – one real and one forged – and the resulting outputs would be measured. The success of the experiment would be measured by the error in the training process and the success of the testing stage. After measuring the results, they would be compared to results from other studies on image recognition, specifically, signature recognition, and a conclusion would be drawn afterwards.

## Input

Firstly, images or scans of the signatures have to be taken in order to generate input. All of the images will be recorded in similar lighting conditions on the same paper, and the signatures will be written with the same pen to minimise external factors that might influence the results and to emulate as closely as possible the similarity of the background for digitally written signatures, as can be seen in figure A.

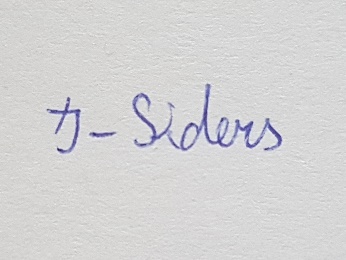
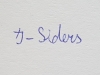


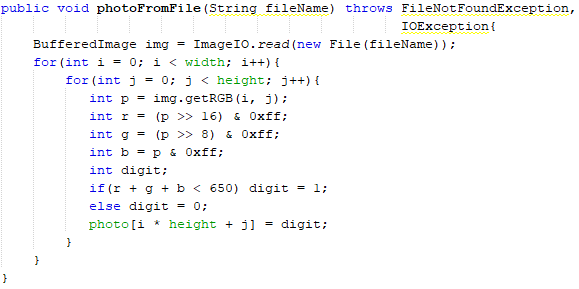
Figure B

Figure A

Figure C

Secondly, the images have to be converted to input for the network to analyse. This can be accomplished by compressing the images in this specific system to a smaller size (75 \* 100 pixels) using an Internet resource[[3]](#footnote-3), and the paths to the files will be given as properties of the image class, as can be seen in figure B.

A file with a photograph extension (like .png, .jpeg or other) cannot be properly analysed in a simple neural network, so to solve the problem the photograph of the signature is converted into an array of ones and zeroes (as many as there are pixels, shown in figure C as a small fragment of the whole) so that each pixel is represented in the input. To convert the picture into an array, each pixel’s colour is analysed, and if the red, green, and blue levels are below 650, the pixel is associated with the digit 1, if the levels do not exceed 570, the digit is 0. The code for it is shown below:



One loss from this conversion is that similarly shaded pixels end either as the same digit or the complete opposite, resulting in a very lossy translation from a multi-variable system (combinations of red, green, blue) to a binary system (ones and zeroes), which means that possibly valuable data is discarded. This affects mostly the edges of a signature, but in electronically written ones the edges are sudden and do not grade. Therefore, one further step is taken to emulate passport security signatures.

## Construction and Calculation

The proceeding action is the construction of the artificial neural network itself. A neural network consists of perceptrons (another name for neurons in an artificial neural network), each of which has weights, biases and outputs from previous perceptrons assigned to them. A weight is, in essence, the importance of the perceptron’s output in relation to other outputs[[4]](#footnote-4). The weights at first have random numbers assigned because the network has not been trained yet, so that it can have something to check and improve. A bias is a value that is also weighted and then applied to every perceptron to react according to the conditions of the network. The neural network is assembled in layers where each perceptron is connected to all perceptrons in the previous layer. The layers are divided into three self-evidently named categories – input, output and hidden layers. In this case, the input layer will consist of 7500 perceptrons, each holding the value of either 1 or 0, depending on the pixel’s colour in that specific position. The hidden layer count and sizes are very much dependent on the type of neural network required, and image recognition networks need at least 2 hidden layers to go through. The exact number of perceptrons and layers needed cannot be calculated; therefore, many presumptions have to be made, finally resulting in 2 hidden layers, each having 75 perceptrons in them. The output layer will consist of only 1 perceptron since the network will be built for the purpose of answering a simple “yes” (1) or “no” (0) question regarding the genuineness of the input signatures, the final layer’s perceptron’s output value being the answer.

The raw output of a perceptron is calculated by the sum of inputs of perceptrons in the previous layer which are first multiplied by their respective weights plus an added bias, which itself is multiplied by a weight. As an equation it would look something like this: .

And a simplified version of the code would look like this:



## Normalisation by an Activation Function

If the range of a perceptron’s possible output values were infinite, there would be considerable difficulties in the analysis of the output and sometimes errors could occur. The solution to this problem is once again an imitation of the human brain in the way synapses fire. In the brain, the electrical impulses in neurons must meet a certain threshold to fire, and the perceptrons in artificial neural networks work in a similar fashion. To simulate the binary system in biology and to normalise, i.e., express in some standard form, the output, an activation function is added to the output. In this case, the sigmoid function will be used because it is one of the simplest and easiest functions to understand and analyse. The sigmoid function (represented mathematically as ) changes the range to numbers between 0 and 1 and utilises a simple slope (as can be seen in Figure D) which will be used later in the training process.



Figure D[[5]](#footnote-5)

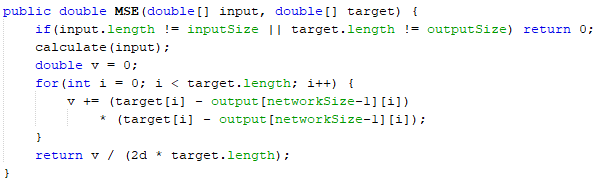
The action of creating a sum from the previous layer’s weighted outputs and applying an activation function to it is called forward propagation.

# Training the Network

Once the structure is created and the inputs are imported, the weights of the perceptrons still have not changed and are still the same random values. For the network to learn, the weights must be changed in such a way that the errors between the expected outputs and the actual outputs would be minimised. In supervised learning, neural networks are not taught from the beginning in the same way that children are taught by teachers and parents, but the acquisition of information is accomplished by learning from mistakes (these mistakes can be made only by the network trying randomly, which is why the weights are random at the start). This “learning” is achieved by backpropagation.

## The Cost Function

To understand backpropagation, one must first start with the cost function (also called the loss function). Once the outputs have been calculated with random weights, they can be compared to the values of expected outputs, that is, the so-called target outputs. The cost function is simply a measure of how poor the network’s performance is. The bigger the value, the poorer the performance. If the value is close to zero, the cost of the network is almost nothing, which means that the deep network is doing well. Neural networks can greatly vary in the choice of a cost function (because there are many varieties used for different types of neural networks) and in this instance, the deep neural network will be using a mean squared error (which usually is an average of many mean squared errors), the mathematical equation for the function being and the code looking like this:



## Backpropagation

Backpropagation is an algorithm which provides detailed insights into how changing the weights and biases affects the overall behaviour of the network. Once the cost function is calculated, the main goal of the network is to minimise the value of the function. In a sense, backpropagation and the training of the whole neural network is “just minimising a cost function” (Sanderson [[6]](#footnote-6)). This is achieved by a method called gradient descent.

### Gradient Descent

To code a way for the purpose of getting the local minima of a function (the minimum value of the cost function), the world of computer science once again makes use of calculus. Gradient descent is an algorithm which finds the gradient of a certain function (in this case, a loss function) and then changes variables (in this case, weights) according to the type and steepness of the gradient. To better understand gradient descent, many computer science teachers give the example of an imaginary ball rolling down in a parabola (one of the simplest of functions with a minimum value). If the gradient of the ball’s position on the parabola is positive (the ball stands on the right-side branch), the weights should be adjusted with a negative number, i.e., the value of the cost function would decrease if the ball rolled left and, therefore, down. The opposite goes for the situation where the gradient of the ball’s position on the parabola is negative. Since the descent is multiplied by the gradient, the cost function is minimised not only in the right direction, but also always a little bit closer to the local minimum, somewhat resembling the race of Achilles paradox as he always travels half the distance to the finish line. Each iterative step to the local minimum is smaller because the distance is smaller until finally the computer approximates the step’s size to be 0[[7]](#footnote-7)[[8]](#footnote-8).

## Solutions to Internal Problems

Training a deep neural network is not a simple task; therefore, some problems in the training phase of the system have to be solved. For example, pictures of the signatures have to be compressed into smaller sizes to make the input size manageable for the computer in charge of computing the results of the experiment. This compression can cause a loss of detail in the process. However, firstly, that detail would also be missing in a digitally written signature, e.g., for passport security, because of the lack of additional variables like the strength of the pushing of the pen and the length of holding it in this method, and, secondly, computers at the disposal of the government/security agencies would have no need for this compression because of the greater computing power at their disposal. Therefore, the loss of information would be negligible, too. Furthermore, another problem encountered in this experiment is when the last signature is used in the training process, and the output from the testing images tends towards the target of that last signature. One possible solution to this problem is to use the first signatures in the training cycle more and then gradually lessen the impact each consecutive image has on the neurons of the network, which leads to the improved reliability of the resulting outputs. The mentioned problems which can occur in the training phase are one of the reasons why neural networks could be considered as ineffective signature verifiers as of yet.

# Testing the Network and Results

Once the network was built and ready, and the neurons in it were trained to recognise forged signatures from the genuine ones, it was time to test it and get the results needed. The network was tested by inputting one image of an authentic signature and one image of a forged one without their targets (their desired outputs would otherwise already tell the network the status of the signatures). From each signature the neural network outputs the value of the images using its newly-trained neurons and the final mean squared errors, which were the largest in the testing phase because they could not be changed by the network as was the case in the training phase.

Figure E

As can be seen in Figure E, the output from real signatures (represented by the blue columns) was, at its maximum, 0.98, when its target was 1. Additionally, the output from forged signatures (represented by the orange columns) was, at its minimum, 0.13, when its target was 0. The third pair of authentic and forged tested images was outside the norm of the other tests either because of the irregularity of the genuine signature’s representation of the signatory’s usual signature, i.e., genuine signatures can sometimes differ from normal, or because of the above average quality of the forgery.

The experiment was then improved upon by making changes to the acquiring of the photos of signatures, firstly, by scanning them instead of taking a picture and, secondly, by immediately converting them into black and white instead of dealing with the ambiguity of paper colour or the problem of which is the exact edge of the ink of the signature. The second signature was chosen as a different type of signature – with a few specific strokes (5), creating a sort of stamp or symbol, whereas the first one was a word in the handwriting of that person. To also contrast the first set of signatures, these were written in a box, such as one that could be found in passport security, to see if this limitation affected the accuracy of results in any way.

**Second set of signatures, written in a box, in black and white**

Forged signature

Genuine signature



The results of the second set of signatures were then compiled in Figure F below.

Figure F

Judging from the results of the second set of signatures, it can be concluded that a neural network can recognise that a genuine signature is indeed a signature, but it fluctuates when it is given a forged one – possibly because the quality of the forgery differs.

Most importantly, the difference between the outputs of each pair of signatures can be seen, and the trend of each output towards their respective targets is noticeable; therefore, the making of a deep neural network that could distinguish between authentic and forged signatures could be considered a success. However, the hypothesis proved to be correct not only in that regard, as there was a meaningful error to be considered.

# Analysis of the Results and Errors

The error in this experiment was measured by determining the mean squared error, which was calculated both during the training process and in the last phase of the experiment, the testing process.

Figure F

As can be seen in Figure E, the mean squared error ranges from about 0.6% to 6.9%. The error of the experiment varies from considerably small to understandably large values compared to that of errors in other image recognition neural networks, e.g., 3.2%[[9]](#footnote-9) and 1.24% (with the rejection rate of 4.506% and reliability factor of 0.987%)[[10]](#footnote-10). There are many factors which could influence the error significantly when trying to verify if a signature is authentic or forged. For example, the signature in the box could be shifted in any direction (up, down, left, right), which the neural network considers an entirely new signature with different positions of its pixels and not just the same signature but just shifted. Convolutional neural networks (CNNs) are neural networks which take into account the shiftability of objects in images and recognise the same object but in a different place in an image. Additionally, CNNs work with a special system of numerous filters that can be applied to an image by methods such as pooling to acquire different views of it, e.g., to highlight edges or colour differences[[11]](#footnote-11); thus, they would be able to analyse different aspects of signatures to learn about them bit by bit. Therefore, less complex neural networks (even deep networks) are incapable of truly verifying the genuiness of signatures. Furthermore, the emotional and physical state of the signatory may vary at different times of writing his or her name, since human handwriting differs depending on whether they are distressed, happy, angry or in any other mood, which influences muscular tension, thereby influencing handwriting[[12]](#footnote-12), which consists of many different variables, such as slant, height and width. The context in which the signatory affixed his or her signature is also important, for example, caffeine is shown to influence handwriting by improving the signatories’ writing speed and fluency[[13]](#footnote-13); therefore, the handwriting and, therefore, signature might be different if the signatory drinks a caffeine-full drink, such as an energy drink or regular coffee, before signing his or her signature. All of these examples provide meaningful variables in the context of signature-writing, which can only be analysed by human experts, i.e., graphologists, when deciding if a signature is genuine or forged. When not considering these outside factors (*ceteris paribus*), neural networks (and especially CNNs) are capable of determining the reliability of a signature to some extent.

Figure G

Figure F displays the outputs of each pair of signatures with the average mean squared error also shown. The lack of target outputs inside the error bars means that the neural network did not calculate the outputs to the best of its abilities, which can be seen as a possible future improvement to the network.

# Conclusion

The hypothesis was true to some extent: the neural network did distinguish authentic signatures from forged ones; however, it did so with a much smaller error than anticipated. The network correctly labelled each tested signature and was always right in the value tending towards a 1 or a 0 (in the first set of signatures) even if it was close to the 0.5 edge (with one outlier in the second set of signatures).

Convolutional neural networks could, therefore, be used to verify passport signatures in less important matters, but experts should still handle highly important cases personally because of the current limitations of neural networks when it comes to signatures. The largest limitations are those of human psychology and context, since the neural networks do not yet consider the reasons behind the characteristics of people’s handwriting and how it is affected by the personality of the signatory. The detailed aspects of a signature, like the combination of slant, amplitude and size, are determined by the character of the signatory and other variables unobservable to the neural networks. Additionally, the context is also very important when determining whether the signatory was overwhelmed by any particular emotion or under influence of any specific physical state, signals received by the system of perception when affixing his or her signature.

# Works Cited

Drouhard, J.-P., Sabourin, R., Godbout, M. “A Neural Network Approach to off-Line Signature Verification Using Directional PDF.” Pattern Recognition, vol. 29, no. 3, 1996, pp. 421., doi:10.1016/0031-3203(95)00092-5.

Eggers, Finn. NN – Fully Connected Tutorial (videos). *YouTube*, 31 January 2018, <https://www.youtube.com/playlist?list=PLgomWLYGNl1dL1Qsmgumhcg4HOcWZMd3k>. Accessed 15 August 2018.

Free Online Image Resizer and Converter, <https://www.fixpicture.org/>. Accessed 12 November 2018.

Johnson, Erica. “Document forgery in financial industry more common than you'd think, past employees say.” *CBC*, 31 May 2017, [https://www.cbc.ca/news/business/financial-industry-employees-forge-documents-more-often-than-you-d-think-1.4138212. Accessed 11 November 2018](https://www.cbc.ca/news/business/financial-industry-employees-forge-documents-more-often-than-you-d-think-1.4138212.%20Accessed%2011%20November%202018).

Karpathy, Andrej. “CS231n Convolutional Neural Networks for Visual Recognition” Stanford study notes. Spring 2018, <http://cs231n.github.io/convolutional-networks/>. Last accessed 1 March 2019.

Naftali, A. "Behavior Factors In Handwriting Identification". The Journal Of Criminal Law, Criminology, And Police Science, vol 56, no. 4, 1965, p. 532. *JSTOR*, doi:10.2307/1141688.

Nam, Seungsoo, et al. “Forged Signature Distinction Using Convolutional Neural Network for Feature Extraction.” Applied Sciences, vol. 8, no. 2, 2018, p. 10., doi:10.3390/app8020153.

Nielsen, Michael. “Neural Networks and Deep Learning”. 25 November 2013, <http://neuralnetworksanddeeplearning.com/>. Accessed 12 Nov 2018.

3Blue1Brown (Sanderson, Grant). Gradient descent, how neural networks learn | Deep learning, chapter 2 (video). *YouTube*, 16 October 2017, <https://www.youtube.com/watch?v=IHZwWFHWa-w>. Accessed 15 August 2018.

Shiffman, Daniel. “The Nature of Code”. 13 December 2012, <http://www.natureofcode.com/book>. Accessed 12 Nov 2018.

“Sigmoid Function.” *Wikipedia*, Wikimedia Foundation, 22 Dec. 2018, <https://en.wikipedia.org/wiki/Sigmoid_function>. Accessed 3 Jan. 2019.

Tucha, Oliver, et al. “The Effect of Caffeine on Handwriting Movements in Skilled Writers.” Human Movement Science, vol. 25, no. 4-5, 2006, pp. 523–535., doi:10.1016/j.humov.2006.06.001.

# Appendix

**Results table of first set of signatures**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Output | | Mean Squared Error | | | |
| Real | **Forged** | **Real** | **Forged** | **Worst** | **Average** |
| 0,78 | 0,18 | 0,0239 | 0,0170 | 0,0239 | 0,0205 |
| 0,86 | 0,13 | 0,0100 | 0,0080 | 0,0100 | 0,0090 |
| 0,57 | 0,29 | 0,0935 | 0,0441 | 0,0935 | 0,0688 |
| 0,85 | 0,16 | 0,0101 | 0,0123 | 0,0123 | 0,0112 |
| 0,98 | 0,16 | 0,0002 | 0,0129 | 0,0129 | 0,0066 |
| 0,92 | 0,24 | 0,0028 | 0,0286 | 0,0286 | 0,0157 |
| 0,91 | 0,12 | 0,0044 | 0,0073 | 0,0073 | 0,0059 |

**Results table of second set of signatures**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Output | | Mean Square Error | | | |
| Real | **Forged** | **Real** | **Forged** | **Worst** | **Average** |
| 0.73 | 0.18 | 0.0378 | 0.0167 | 0.0378 | 0.02725 |
| 0.75 | 0.02 | 0.0313 | 0.0003 | 0.0313 | 0.01580 |
| 0.97 | 0.20 | 0.0006 | 0.0209 | 0.0209 | 0.01075 |
| 0.94 | 0.09 | 0.0020 | 0.0043 | 0.0043 | 0.00315 |
| 0.90 | 0.26 | 0.0054 | 0.0337 | 0.0337 | 0.01955 |
| 0.82 | 0.03 | 0.0170 | 0.0004 | 0.017 | 0.00870 |
| 0.88 | 0.04 | 0.0070 | 0.0009 | 0.007 | 0.00395 |
| 0.99 | 0.37 | 0.0001 | 0.0692 | 0.0692 | 0.03465 |
| 0.89 | 0.58 | 0.0062 | 0.1654 | 0.1654 | 0.08580 |
| 0.93 | 0.17 | 0.0024 | 0.0149 | 0.0149 | 0.00865 |

## Signature class file

package eesignatures;

import java.awt.image.BufferedImage;

import java.io.File;

import java.io.FileNotFoundException;

import java.io.IOException;

import javax.imageio.ImageIO;

public class EESignatures {

int width = 100;

int height = 75;

int[] photo = new int[7500];

private String loc;

public EESignatures(String location){

loc = location;

}

String getLoc(){

return loc;

}

public void photoFromFile(String fileName) throws FileNotFoundException, IOException{

BufferedImage img = ImageIO.read(new File(fileName));

for(int i = 0; i < width; i++){

for(int j = 0; j < height; j++){

int p = img.getRGB(i, j);

int r = (p >> 16) & 0xff;

int g = (p >> 8) & 0xff;

int b = p & 0xff;

int digit;

if(r+g+b < 650) digit = 1;

else digit = 0;

photo[i \* height + j] = digit;

}

}

}

}

1. Johnson, Erica. “Document forgery in financial industry more common than you'd think, past employees say.” *CBC*, 31 May 2017, <https://www.cbc.ca/news/business/financial-industry-employees-forge-documents-more-often-than-you-d-think-1.4138212>. Accessed 11 November 2018. [↑](#footnote-ref-1)
2. Eggers, Finn. NN – Fully Connected Tutorial (videos). *YouTube*, 31 January 2018, <https://www.youtube.com/playlist?list=PLgomWLYGNl1dL1Qsmgumhcg4HOcWZMd3k>. Accessed 15 August 2018. [↑](#footnote-ref-2)
3. Free Online Image Resizer and Converter, <https://www.fixpicture.org/>. [↑](#footnote-ref-3)
4. Nielsen, Michael. “Neural Networks And Deep Learning”. 25 November 2013, <http://neuralnetworksanddeeplearning.com/>. Accessed 12 Nov 2018. [↑](#footnote-ref-4)
5. “Sigmoid Function.” *Wikipedia*, Wikimedia Foundation, 22 Dec. 2018, <https://en.wikipedia.org/wiki/Sigmoid_function>. Accessed 3 Jan. 2019. [↑](#footnote-ref-5)
6. 3Blue1Brown. Gradient descent, how neural networks learn | Deep learning, chapter 2 (video). *YouTube*, 16 October 2017, <https://www.youtube.com/watch?v=IHZwWFHWa-w>. Accessed 15 August 2018. [↑](#footnote-ref-6)
7. Nielsen, Michael. “Neural Networks And Deep Learning”. 25 November 2013, <http://neuralnetworksanddeeplearning.com/>. Accessed 12 Nov 2018. [↑](#footnote-ref-7)
8. Shiffman, Daniel. “The Nature of Code“. 13 December 2012, <http://www.natureofcode.com/book>. Accessed 12 Nov 2018. [↑](#footnote-ref-8)
9. Nam, Seungsoo, et al. “Forged Signature Distinction Using Convolutional Neural Network for Feature Extraction.” Applied Sciences, vol. 8, no. 2, 2018, p. 10., doi:10.3390/app8020153. [↑](#footnote-ref-9)
10. Drouhard, J.-P., et al. “A Neural Network Approach to off-Line Signature Verification Using Directional PDF.” Pattern Recognition, vol. 29, no. 3, 1996, pp. 421., doi:10.1016/0031-3203(95)00092-5. [↑](#footnote-ref-10)
11. Karpathy, Andrej. “CS231n Convolutional Neural Networks for Visual Recognition” Stanford study notes. Spring 2018, <http://cs231n.github.io/convolutional-networks/>. Last accessed 1 March 2019. [↑](#footnote-ref-11)
12. Naftali, A. "Behavior Factors In Handwriting Identification". The Journal Of Criminal Law, Criminology, And Police Science, vol 56, no. 4, 1965, p. 532. *JSTOR*, doi:10.2307/1141688. [↑](#footnote-ref-12)
13. Tucha, Oliver, et al. “The Effect of Caffeine on Handwriting Movements in Skilled Writers.” Human Movement Science, vol. 25, no. 4-5, 2006, pp. 523–535., doi:10.1016/j.humov.2006.06.001. [↑](#footnote-ref-13)