مجلة
كلية التـــراث الجامعة

مجلة علمية محكمة

متعددة التخصصات نصف سنوية

العدد التاسع والثلاثون

عدد خاص بوقائع المؤتمر العلمي السنوي السادس عشر (الدولي الخامس)

18 نيسان 2024

ISSN 2074-5621

رئيس هيئة التحرير

أ.د. جعفر جابر جواد

مدير التحرير

أ.م. د. حيدر محمود سلمان

رقم الايداع في دار الكتب والوثائق 719 لسنة 2011

مجلة كلية التراث الجامعة معترف بها من قبل وزارة التعليم العالي والبحث العلمي بكتابها المرقم (ب 3059/4) والمؤرخ في (4/7 /2014)

Digital Recognition System Attacks Through Noisy Image Using Machine Learning

Fadhil Abbas Fadhil* / May Sabri Mmohammed*
Noor Muneam Abbas* / Nikolai Safiullin**

* قسم علوم الحاسوب، الجامعة التكنولوجية، بغداد، العراق. ** جامعة الأورال الفيدرالية، كاترينبورغ، روسيا

Abstract
Images with fine imperfections introduced by malicious optimization techniques to deceive classifiers are examples of attacks in computer vision. Machine learning models have proved their susceptibility to attacks from the adversary, despite displaying excellent capability in completing complex cognitive tasks, particularly those linked to computer vision, such as classification and picture recognition. This shortcoming compromises the deployment of these learning algorithms, which could be detrimental to future progress in this area. Initially promising, the various defenses proposed proved brittle and ineffective against powerful and adaptable attacks. Numerous solutions have been proposed in the literature to address these vulnerabilities. However, developing an effective defense mechanism is challenging, as several methods have previously been shown to be ineffective against attackers. That is, a similar task of research in this area is relevant. The purpose of the work: to study the performance of the digit recognition system with noisy images. To solve this problem, you will need first develop the system for recognizing digits in images using one of the machine learning methods; achieve sufficiently high accuracy of this system; noise the images from the training sample and evaluate the final accuracy of the recognition results without retraining the system; process the obtained results and draw conclusions.

Keywords: Image Recognition, Machine Learning, Artificial Neural Networks, Attack Information Systems, Cyber security.

الهجمات التي تستهدف أنظمة التمييز الرقمي عبر الصور المشوشة باستخدام (Machine Learning)

* قسم علوم الحاسوب، الجامعة التكنولوجية، بغداد، العراق. ** جامعة الأورال الفيدرالية، كاترينبورغ، روسيا.

الخلاصة
الصور ذات العيوب الدقيقة التي أدخلتها تقنيات التحسين الخبيثة لخداع المصنفات هي أمثلة على الهجمات في الرؤيا بالحاسوب. أثبتت نماذج التعلم الآلي تأثرها بالهجمات من الخصم، على الرغم من إظهار قدرة ممتازة في إكمال المهام المعرفية المعقدة، لا سيما تلك المرتبطة بالرؤيا بالحاسوب، مثل التصنيف والتعرف على الصور. يؤثر هذا القصور على خوارزميات التعلم، والتي قد تكون ضارة بالتقدم المستقبلي في هذا المجال. واعدة في البداية، أثبتت الدفاعات المختلفة التي اقترحت أنها هشة وغير فعالة ضد الهجمات القوية والقابلة للتكيف. تم اقتراح العديد من الحلول في البحوث لمعالجة نقاط الضعف هذه. ومع ذلك، فإن تطوير آلية دفاع فعالة يمثل تحديًا، حيث ثبت سابقًا أن العديد من الأساليب غير فعالة ضد المهاجمين. وهذا يعني أن مهمة بحث مماثلة في هذا المجال يعد مفيدا. الغرض من هذا العمل: دراسة أداء نظام التعرف على الأرقام مع الصور المشوشة. لحل هذه المشكلة، تم القيام أولاً بتطوير نظام للتعرف على الأرقام في الصور باستخدام إحدى طرق التعلم الآلي؛ مع تحقيق دقة عالية بما فيه الكفاية لهذا النظام؛ إضافة ضوضاء الى الصور من عينة التدريب وتقييم الدقة النهائية لنتائج التعرف دون إعادة تدريب النظام؛ معالجة النتائج التي تم الحصول عليها واستخلاص النتائج.

1. Introduction

Deep learning algorithms have advanced significantly and quickly in recent years in handling a variety of issues related to sophisticated data processing. Natural language processing and voice recognition have made significant progress [1,2], gaming [3], financial market research, fraud and virus detection [4,5], DDoS protection [6], and computer vision are just a few examples. In the area of computer vision, Convolutional Neural Networks (CNN) are at the forefront of deep learning. After that, the performance of CNNs continues to advance because to growing usage of GPUs and environments, and they are currently utilized in a variety of applications including biometric and handwritten character identification, autonomous cars, surveillance systems, and medicine and diagnostics. Modern picture classification methods have been overtaken by machine learning algorithms, which are currently utilized in security-critical applications like biometric identity systems [19], self-driving automobiles and basic operators inside encryption algorithms [18]. Recent research has shown that even algorithms that can outperform human skills are prone to hostile situations.

Handwriting recognition technologies are used in a variety of settings. For instance, historical papers, bank cheques, and letters must be read and saved using handwriting recognition technology. Online tools are also often utilized for handwriting recognition in addition to these examples. There are educational programs to facilitate handwriting recognition on electronic devices such as tablets, especially in the field of education. Handwriting recognition is used to create personalized applications for people with physical or mental disabilities, as well as for young people.

Handwriting recognition algorithms are included in some smartphone applications that we use on a regular basis. The phone's camera can swiftly read and interpret handwritten texts that are nearby. You may search online or translate scanned materials into a variety of languages.

This paper's primary contributions to study the effect of added noise with a normal distribution on the performance of a digit recognition system developed on the basis of machine learning methods.

The rest of this paper is organized into the following categories: The most recent relevant study is discussed in Section 2. Section 3 of the article discussed machine learning. Neural networks are shown in Section 4. The Development of the Recognition System is explained in Section 5. The conclusions and suggestions are offered in Sections 6 and 7.

2. Related Work

The identification of handwriting has been the focus of several investigations. With an updated National Institute of Standards and Technology (MNIST) dataset, Engine Dagdeviren's handwriting recognition study's accuracy was compared with that of SVM and ANN. He used SVM in his MATLAB tests, and on an analysis of 10,000 data points, he had a 99.97% success rate. He developed a handwriting recognition tool using ANN and got an 80.39% accuracy rate on a sample of 10,000 data points [7]. Previously, the work of the attacker was limited to networks and encryption algorithms to obtain data illegally, but recently, with the development of modern technology and the use of artificial intelligence in large and frequent use in our daily lives, especially machine learning, this was included in the plans of the attacker, in addition to the fact that the attacker can also use artificial intelligence to carry out their attack.

The scientific community has been forced to reevaluate all the steps involved in developing creative models, from creating architectures to formulating the learning algorithms used, as a result of CNNs and other deep learning algorithms being vulnerable to adversary attacks. The goal is to hypothesize some potential causes for this lack of robustness and, as a result, suggest countermeasures that can thwart future attacks from the adversary. Adversarial machine learning is a recent research area that, in a nutshell, seeks to develop more reliable deep learning models as a result of a race between both attacking and defending against hostile cases.

3. Machine Learning

Currently, machine learning lags behind some of the most important technological advancements. It is used in the growing industry of self-driving cars, as well as for the study of galaxies, as it helps in the discovery of exoplanets. Stanford University recently described machine learning as "the study of how to make computers work without being explicitly programmed." Machine learning has generated many new ideas and technologies, including unsupervised, supervised learning, algorithms for robots, the Internet of Things, analytics tools, chatbots [8]. Machine learning employs a variety of techniques to address data-related issues. There is nothing known as a universal method for issue resolution, data analysts like to remind out. The sort of issue you are attempting to solve, the number of variables involved, the proper model to apply, and other considerations all influence the strategy that is employed. Here are some of the most commonly used machine learning (ML) algorithms [9]. So, there is more than one type of machine learning.

4. Neural Network

A neural network is a mathematical model that consists of several layers of elements that perform parallel computations, and the inspiration for creating neural networks originally came from research on the information processing mechanisms of the biological nervous system, especially the human brain[10,11] , and a neural network is made up of multiple layers, each of which is divided into three sections: the input layer, the output layer, and a few layers that remain hidden between them, as shown in the image below:

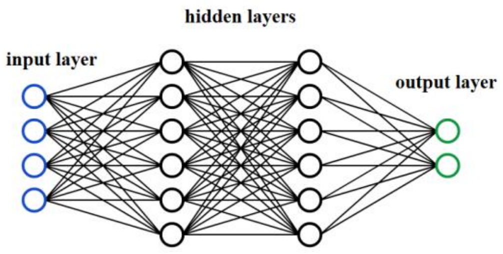


Figure 1: Neural network

The beginning of neural networks dates back to 1943, when neuroscientist Warren McCulloch and mathematician Walter Bates wrote a paper on how neurons work [7]. They created a straightforward neural network models using electrical networks similar to this to explain how neurons function in the brain [12,13].

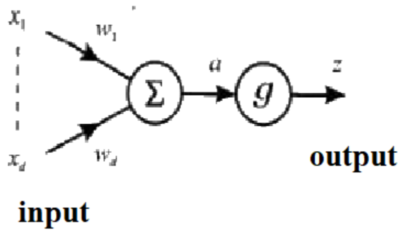


Figure 2: neural network model

A collection of input variables xi(i=1,...,d) are transformed into an output variable z via this model, which may be thought of as a non-linear function. Additionally, the McCulloch-Pitts model's equation:

$$a = \sum_{i=1}^{d} wixi + w0 \ldots (1)$$

The signal x_i at input i is multiplied by a parameter wi, referred to as the weight, before being added to all other weighted inputs to create a common input. The bias parameter w0 is referred to as the bias, and officially the bias can be viewed as a special case of the weight obtained from the extra input, the value whose x_0 is forever set to +1. The equation so appears as follows:

$$a = \sum_{i=1}^{d} wixi \ldots (2)$$

Where x0 = 1. It should be noted that the weights (and bias) can relate to either excitatory or inhibitory synapses and can be of any sign. Then, an operation on a with a non-linear activation function g() yields the output z of unity, which is roughly analogous to the neuron's average fatigue rate, so that

$$z = g(a)$$

5. Development Of the Recognition System

To create a handwriting recognition model, we need a working environment, algorithms, and several important libraries. The TensorFlow library was used. It is a key Python package that is free source and used for quick digital computation [14,15]. Where it can be used or its libraries can be used to build deep learning models, and thanks to its flexible and comprehensive ecosystem, developers and researchers can easily publish and build machine learning applications.

The data was loaded, which was previously available in the Keras library along with other datasets including the CIFAR10, CIFAR100, Boston home price regression, and IMDB movie review rating datasets. The MNIST dataset was loaded into variable objects after being imported from the Keras datasets. Then, using the objects load_data [18] method, we return the test data (test_img), along with its tags (train_lab), and the training data (train_img).

The MNIST dataset contains 60,000 small 28×28 pixel grayscale square images consisting of handwritten single digits from 0 to 9. This is a commonly used and understood dataset that is basically "solved" [20]. Deep Learning Convolutional The most effective models are neural networks, which on the test dataset had a classification accuracy of over 99% and an error rate between 0.4% and 0.2%. The data set contains 60,000 photos, of which 50,000 are provided for training and 10,000 for testing.

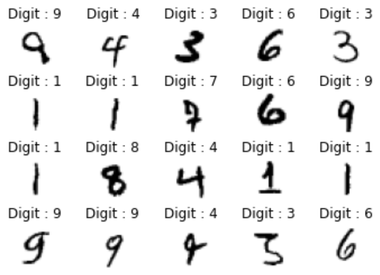


Figure 3: Handwritten numbers with their real meaning

Before transferring the data to the network, we normalize it. Normalizing the input data considerably speeds up the training process. Since we are using stochastic gradient descent, there is less chance of getting stuck in local optima because we are determining the best weights for the network. The values of the pixels vary from 0 to 255. Therefore, by dividing each value by the maximum intensity of 255, we normalize the pixel values from 0 to 1. We will successively extract layers from the network using Sequential(), and we will add layers to the model using add().Since each image is 28 x 28, we use Flatten() to compress the input before passing it to the input layer.
The neurons are then linked together (prior with next) using Dense(). The model is a basic neural network of 512 neurons with two hidden layers. For neurons in hidden layers, the activation function of the linear rectifier unit (ReLU) [16] is used. Compiling is a time-saving procedure. It transforms the basic layer sequence we have set up into a very efficient series of matrix transformations in a way that can run on your GPU or CPU, depending on how Keras is configured.
The optimizer, loss, and metrics are the three foundational elements of the model. The optimizer manages the learning rate. "adam" is the name of the optimizer we employ. This is a wonderful optimizer to utilize, generally speaking. Throughout the training, he regulates the pace of learning. We will choose "Sparse Categorical Cross entropy" as the loss function since it uses one integer per class rather than the complete vector, which saves memory and calculation time. If the score is lower, the model performs better. The "accuracy" measure will be used to evaluate the accuracy score on the set of validation data once the model has been trained in order to establish its correctness. our model is trained with the fit() technique. The three important parameters are training data (train_img), training labels (train_lab), and number of epochs.
The era of machine learning is a full pass of the training data set through the algorithm. For the algorithm, the number of epochs is a critical hyperparameter. It defines the number of epochs (or full passes) in the process of training or training the algorithm for the entire training dataset. The internal model parameters for the dataset change with each epoch. Thus, the batch gradient descent learning algorithm is named after one batch epoch.
Alternatively, it can be thought of as a for loop with a given epoch number, with each path of the loop going through the full set of training data. When using training algorithms, the number of epochs can reach thousands, and the process is programmed to continue until the model error is appropriately minimized.
When building the model, the loss and any other specific metrics are calculated to see if the model is an ideal fit for the given problem and the associated data. This is done using the Model evaluate () method. Using past and present data, predictive modeling is a statistical technique that employs machine learning and data mining to forecast likely future occurrences. It functions by looking at both recent and historical data, then extrapolating that understanding into a model created to forecast future occurrences. Argmax () delivers the indices of the highest values along the axis and is used to forecast the test set by model predict (). As stated earlier, the dataset that is the input to the developed model contains a series of images (in the form of arrays) and each image contains additional information called (actual value), where the actual value is the value number inside the image. We can determine if a machine learning model is performing well based on the image and the actual value.

During the testing phase of the machine learning model, about 10,000 number samples were tested (1000 samples for each number), so when an image and an actual value are entered into the model, the output is a number (n) if that number (n) is equal to the actual value. As a consequence, the model can determine numbers from photographs; however, if the result is not equal, the model was unable to detect that image.
To make it clearer, for the entered numbers (10,000 samples), a model detection table was generated, so we got a table with a size of 10 * 10. (Columns represent the input images, and rows represent the test results from the model).
As we mentioned earlier, when training the model, the number of training times is determined by epochs, and in the first experiment, when epoch = 1, Table1 is obtained.

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
0	979	0	0	0	2	5	3	1	8	2
1	0	984	1	2	0	0	0	0	7	6
2	0	10	958	16	1	1	0	3	9	2
3	0	1	7	955	0	17	0	4	5	11
4	0	2	2	0	951	5	3	0	1	36
5	2	0	1	8	0	984	1	0	2	2
6	6	2	1	0	1	14	971	0	5	0
7	0	7	4	1	4	3	0	907	1	73
8	2	8	2	2	0	15	4	1	960	6
9	2	1	0	6	1	0	2	3	1	984

Table 1: Model test result when epoch = 1

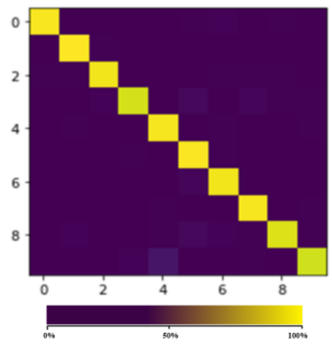


Figure 4: Model test result when epoch = 1

And upon reaching the tenth experience, the numbers were found by 90%, and this is considered a good result. Thus, the final result of testing the constructed model is presented in Table2.

	0	1	2	3	4	5	6	7	8	9
0	971	0	2	0	1	5	11	0	8	2
1	1	983	1	1	4	0	1	1	7	1
2	2	7	957	20	2	1	1	3	6	1
3	0	1	4	976	0	6	3	3	3	4
4	0	0	2	2	985	1	3	0	0	7
5	1	0	1	4	0	986	6	0	2	0

6	1	2	0	0	2	2	990	0	3	0
7	0	2	0	6	6	2	0	975	2	7
8	1	2	0	1	1	3	4	3	983	2
9	1	0	0	5	4	3	1	4	5	977

Table 2: Model test result when number of epochs = 10

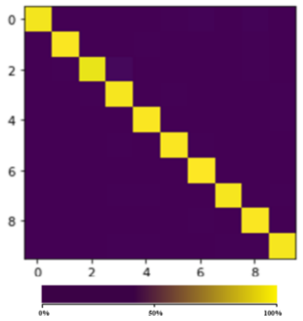
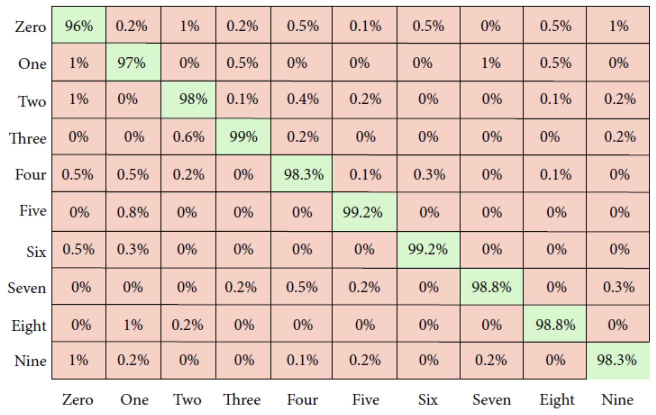


Figure 5: Model test result when number of epochs = 10

The results were compared to a research paper by Syed Sohail, who tried to obtain the best results by using a new technique to recognize handwritten numbers. Syed obtained the following results [17]:



	Zero	One	Two	Three	Four	Five	Six	Seven	Eight	Nine
Zero	96%	0.2%	1%	0.2%	0.5%	0.1%	0.5%	0%	0.5%	1%
One	1%	97%	0%	0.5%	0%	0%	0%	1%	0.5%	0%
Two	1%	0%	98%	0.1%	0.4%	0.2%	0%	0%	0.1%	0.2%
Three	0%	0%	0.6%	99%	0.2%	0%	0%	0%	0%	0.2%
Four	0.5%	0.5%	0.2%	0%	98.3%	0.1%	0.3%	0%	0.1%	0%
Five	0%	0.8%	0%	0%	0%	99.2%	0%	0%	0%	0%
Six	0.5%	0.3%	0%	0%	0%	0%	99.2%	0%	0%	0%
Seven	0%	0%	0%	0.2%	0.5%	0.2%	0%	98.8%	0%	0.3%
Eight	0%	1%	0.2%	0%	0%	0%	0%	0%	98.8%	0%
Nine	1%	0.2%	0%	0%	0.1%	0.2%	0%	0.2%	0%	98.3%

Figure 6: Confusion matrix of the proposed approach for HDR [17]

In Table 3, we can clearly see the improvement that occurred in the model as the number of training epochs increased.

Table 3: Accuracy Ratio Results

	Epochs=1	Epochs=5	Epochs=10
Accuracy	0.9633	0.9766	0.9783
Precision	0.9878	0.988	0.9928
Recall function	0.979	0.988	0.971
f1-score function	0.9834	0.984	0.9817

5.1 Image Noise for Reducing the Performance of the Recognition System

To analyze the strength and security of previously tested machine learning, we had to attack this model with input data. Because photos are used as input to machine learning, an attacker can take advantage of this feature and manipulate the images that are fed into the model throughout the testing phase.

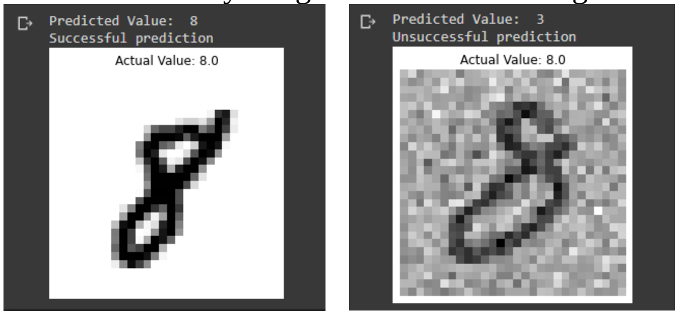
Adding noise to an image is the most popular form of image modification. This entails inserting unwanted data into photos that will affect the image, or trying to change its values to other values, which will lead to a malicious procedure by an attacker to sabotage or reduce the effectiveness of machine learning.
Image noise is the unpredictability of the brightness or color information in the collected photographs. It is an external source that causes image signal degradation. The main thing is that noise is the easiest way to damage images, which can lead to the inability to recognize images.
The system (for failure in its operability), there are many types of noise that do this work. The process of adding noise to the image was used randomly depending on the image pixel value for each point. The process of adding noise to each image is controlled by the Noise Amplitude parameter. Since during training the image pixels were normalized in the range from 0 to 1, the amplitude of the added additive noise varied from 0 to 1, or, more precisely, in fractions (percentages) of the maximum amplitude of the pixels in the images.

$$\mathrm{NI} = \mathrm{I} + (\mathrm{V} * \mathrm{R}) \quad (7)$$

Where:
NI - noisy image.
I - original image.
V – noise amplitude (from 0 to 1).
R – noise with normal distribution.

6. Results and Discussion

The image is displayed and checked (Actual value) to see if it matches the specified number using the Python language function (CHECKING). Although the Actual value should be 8, we note that the noisy image was read as 3 in Figure 6.

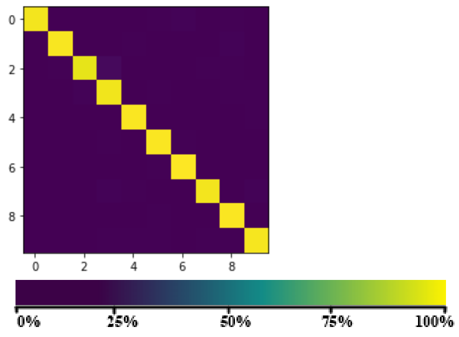
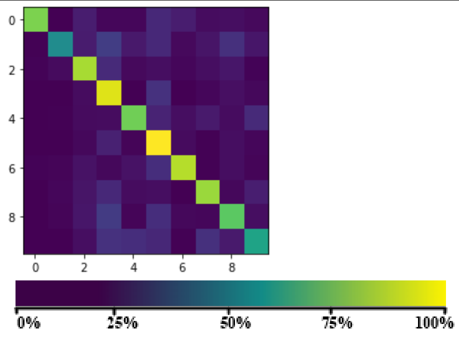
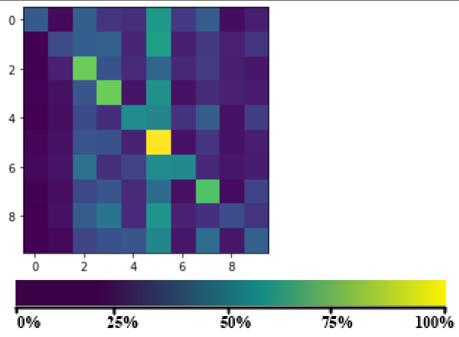


```
Predicted Value:  8
Successful prediction
```

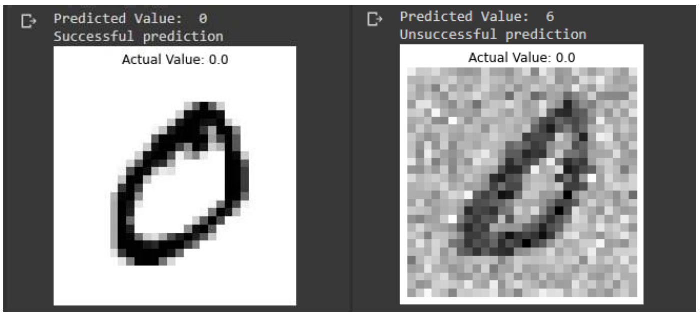
```
Predicted Value:  3
Unsuccessful prediction
```

Figure 7: (a) image number 8 before adding noise, (b) Image number 8 after adding noise.
Using the "Noise Amplitude" factor, 10 experiments were undertaken to attack the machine learning model starting from 10% to 100% to get the real results of the preliminary results table.
The following results show the extent to which the input data was affected after adding noise to it, as we see a clear change in the images and the impact on the result table:

Table 4: Effects of input data after adding noise.

results table	results table image	Noise
[[971 0 3 0 2 5 10 1 6 2] [0 980 1 1 6 0 0 2 9 1] [2 6 953 24 2 1 1 4 7 0] [0 1 8 969 0 7 3 3 5 4] [0 1 2 1 985 1 3 0 0 7] [1 1 1 4 0 982 7 0 2 2] [1 2 0 0 1 5 989 0 2 0] [0 2 1 9 4 1 0 972 2 9] [0 1 0 1 2 5 3 3 982 3] [0 0 0 5 5 4 0 4 6 976]]		**10%**
[[659 1 67 16 15 94 67 28 31 22] [4 399 61 149 57 95 22 48 115 50] [8 24 715 97 21 31 16 30 50 8] [0 3 28 782 3 116 1 14 35 18] [3 6 23 28 644 82 29 60 27 98] [3 2 18 74 11 821 23 6 29 13] [7 11 40 19 39 109 730 5 30 10] [0 13 46 95 26 32 2 698 19 69] [2 11 51 144 12 107 18 13 613 29] [0 1 29 116 106 92 6 115 58 477]]		**50%**
[[141 15 149 76 68 261 83 143 20 44] [5 114 148 152 53 278 43 83 48 76] [6 47 378 125 60 161 58 87 43 35] [8 24 133 378 30 250 27 63 48 39] [3 23 111 66 238 223 74 145 26 91] [12 27 126 124 51 487 26 77 26 44] [14 31 184 69 101 239 231 59 32 40] [6 23 107 131 61 175 24 354 19 100] [4 26 146 192 61 258 49 70 117 77] [3 14 105 125 131 226 34 176 34 152]]		**100%**

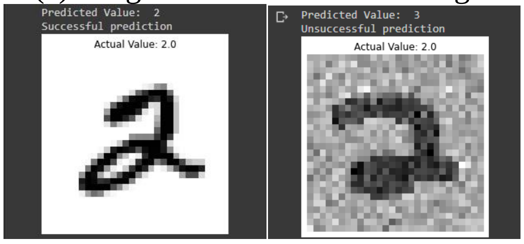
It is clear from the table4 that adding noise with an amplitude of 10%-30% of the maximum amplitude of image pixels reduces the recognition efficiency to about 80%, but still makes recognition a completely possible procedure.
When adding noise with an amplitude of 40%-60% of the maximum amplitude of the image pixels, we notice that the model has suffered and the recognition efficiency has decreased to about 50-60%, which can already be considered a complete violation of the recognition system.
When noise with an amplitude of 70% or more of the maximum amplitude of the image pixels is added, it becomes clear that the model has been completely destroyed and that the trained system has become unusable because it already identifies incorrect data, for example, we note in Figures 7 and 8 when experimenting with the number (0) will give the result (6), which is actually the wrong result.



```
Predicted Value:  0
Successful prediction
Actual Value: 0.0
```

```
Predicted Value:  6
Unsuccessful prediction
Actual Value: 0.0
```

Figure 8: (a) image number 0 before adding noise, (b) Image number 0 after adding noise.

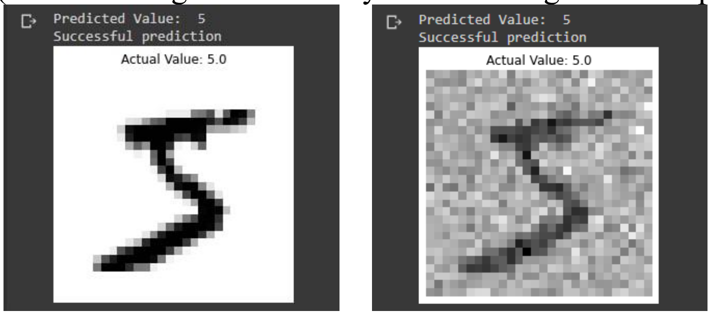


```
Predicted Value:  2
Successful prediction
Actual Value: 2.0
```

```
Predicted Value:  3
Unsuccessful prediction
Actual Value: 2.0
```

Figure 9: (a) image number 2 before adding noise, (b) Image number 2 after adding noise.

But at the same time, we notice (Fig. 9) that some numbers are more stable for recognition than others (“5” was recognized correctly as “5” for high noise amplitudes).



```
Predicted Value:  5
Successful prediction
Actual Value: 5.0
```

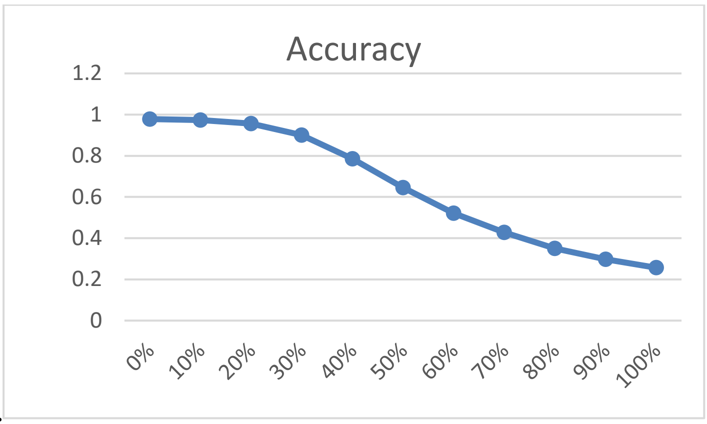
```
Predicted Value:  5
Successful prediction
Actual Value: 5.0
```

Figure 10: (a) image number 5 before adding noise, (b) Image number 5 after adding noise.

To better explain the results, we present the numerical results of the accuracy assessment mentioned earlier in part (3), which show the degree of change that occurs with these measurements before and after adding noise to the model input.

Table 5: How much Accuracy changes when noise is added.

№	Noise	Accuracy
	0%	0.9783
	10%	0.9741
	20%	0.9567
	30%	0.9001
	40%	0.7849
	50%	0.6459
	60%	0.5221
	70%	0.4287
	80%	0.3511
.	90%	0.2982
.	100%	0.2567



.
Figure 11: Plot showing the change in Accuracy when noise is added.

Table 6: The amount of change in Precision when noise is added.

№	Noise	Precision
	0%	0.992843
	10%	0.995881
	20%	0.997912
	30%	0.986052
	40%	0.974699
	50%	0.970972
	60%	0.911392
	70%	0.867612
	80%	0.765043
.	90%	0.688406
.	100%	0.625483

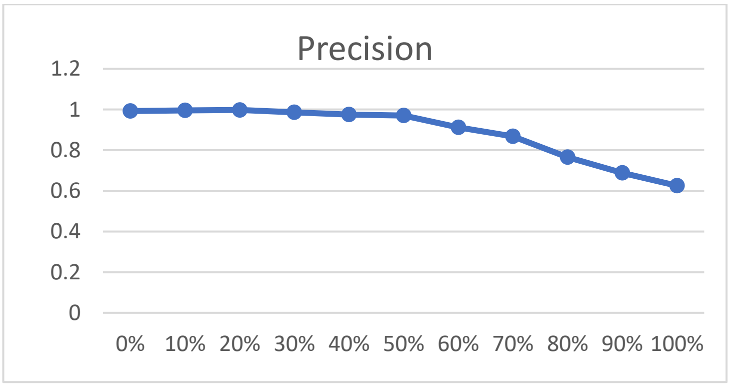


Figure 12: Plot showing the change in Precision when noise is added.

Table 7: How much Recall changes when noise is added.

№	Noise	Recall
	0%	0.971
	10%	0.967
	20%	0.956
	30%	0.919

	40%	0.809
	50%	0.669
	60%	0.504
	70%	0.367
	80%	0.267
.	90%	0.19
.	100%	0.162

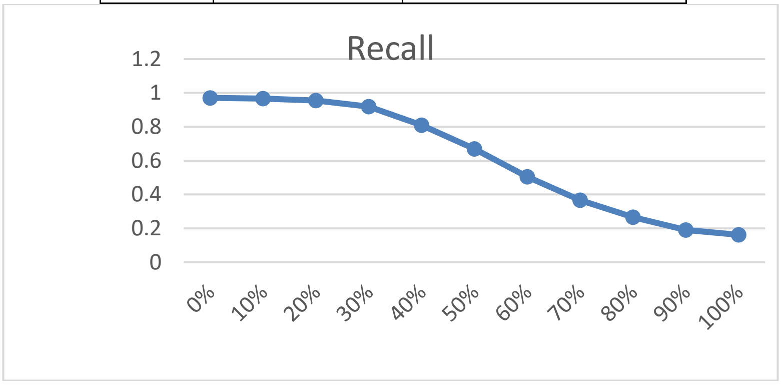


Figure 13: Plot showing the change in Recall with the addition of noise.

Table 8: Rate of change in F1_score when noise is added.

№	Noise	F1_score
	0%	0.9818
	10%	0.981228
	20%	0.976507
	30%	0.951346
	40%	0.884153
	50%	0.792185
	60%	0.649066
	70%	0.515812
	80%	0.395849
.	90%	0.297806
.	100%	0.257347

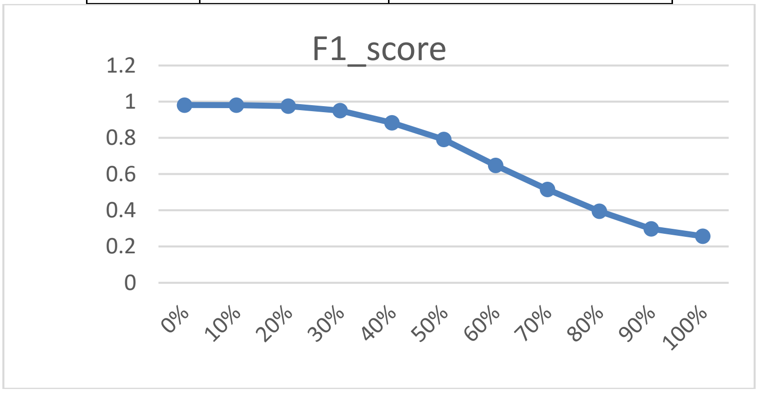
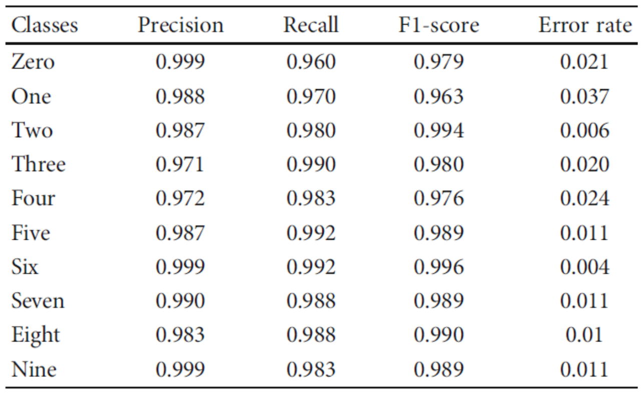


Figure 14: Graph showing the change in F1_score when adding noise.

Syed received category-level performance results for EfficientDet-D4 for HDR recognition [17]:

Table 9: Class-wise performance results of the EfficientDet-D4 for HDR recognition



Classes	Precision	Recall	F1-score	Error rate
Zero	0.999	0.960	0.979	0.021
One	0.988	0.970	0.963	0.037
Two	0.987	0.980	0.994	0.006
Three	0.971	0.990	0.980	0.020
Four	0.972	0.983	0.976	0.024
Five	0.987	0.992	0.989	0.011
Six	0.999	0.992	0.996	0.004
Seven	0.990	0.988	0.989	0.011
Eight	0.983	0.988	0.990	0.01
Nine	0.999	0.983	0.989	0.011

In this study, the machine learning technique, which is based on an artificial neural network, was used to recognize handwritten digits. A collection of handwritten numbers from MNIST were tested. The following conclusions were reached after carefully examining the results:

1. As the amplitude of the noise increases, the recognizable values of the digits begin to gather into some stable groups. That is, when errors occur in recognition, the trained system leaves only a few stable states.
2. In the test part, after adding high noise, we get the following results of the final "groups" for each number:

- 0 → {2 (16%), 5 (27%), 7 (13.5%)};
- 1→ {2 (13%), 3 (17.6%), 5 (25%), 7 (8.5%), 9 (7.3%)};
- 2→ {3 (14.3%), 5 (15.6%)};
- 3→ {2 (13.7%), 5 (24.9%), 7 (6.4%)};
- 4→ {2 (11%), 5 (22.2%), 7 (16.6%), 9 (9%)};
- 5→ {2 (10,7%), 3 (15%), 7 (7.3%)};
- 6→ {2 (16.9%), 4 (8.4%), 5 (26%), 7 (9.2%)};
- 7→ {2 (11.3%), 3 (13.7%), 5 (18.4%), 9 (8.7%)};
- 8→ {2 (16.9%), 3 (19%), 5 (25.2%)};
- 9→ {2 (9%), 3 (11.6%), 4 (12.4%), 5 (22%), 7 (17.7%)}.

3. You can notice that the most common group of numbers.
   This means that for a given trained model system for recognizing numbers in images, this set of numbers should be considered the most likely attack direction, converting into which the system will make mistakes much more often.

7. Conclusion

Thus, the research's objective to evaluate the performance of the digit recognition system in the presence of noisy images, when one of the potential angles of attack on such machine learning systems, was successfully achieved in this final qualification work. To achieve this goal, it was necessary to first develop the system for recognizing numbers in images based on an artificial neural network in two layers. Then this system was trained on data from MNIST (10000 images of numbers), and successfully passed the validation, allowing to achieve an accuracy score of 0.9783. Subsequently, images from the training set were fed to the input of the trained system, but with added noise of different amplitudes (from 0% to 100% of the maximum pixel amplitude), with a further assessment of the recognition accuracy. It is concluded that as the amplitude of the noise increases, the recognizable values of the digits begin to gather into some stable groups. That is, when errors occur in recognition, the trained system leaves only a few stable states.

References

1. Noor Muneam Abbas, Matheel E. Abdulmunim, “mRNA Approach Image Encryption Using LUC Algorithm”, Iraqi Journal of Science, vol. 64, no. 5, pp. 2545–2560, May 2023.
2. Xu, Boyan, et al. "NADAQ: natural language database querying based on deep learning." IEEE Access 7 (2019): 35012-35017.
3. Silver, David, et al. "A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play." Science 362.6419 (2018): 1140-1144.
4. Dahl, George E., et al. "Large-scale malware classification using random projections and neural networks." 2013 IEEE International Conference on Acoustics, Speech and Signal Processing. IEEE, 2013.
5. Erick Knorr. 2015. How PayPal beats the bad guys with machine learning. InfoWorld (Apr 2015). https://www. infoworld.com/article/2907877/machine-learning/how-paypal-reduces-fraud-with-machine-learning.html
6. Yuan, Xiaoyong, Chuanhuang Li, and Xiaolin Li. "DeepDefense: identifying DDoS attack via deep learning." 2017 IEEE International Conference on Smart Computing (SMARTCOMP). IEEE, 2017.
7. Dagdeviren, E., “El yazisi rakam tanima icin destek vektor makinelerinin ve yapay sinir aglarinin karsilastirmasi,” Istanbul University, Master’s Thesis, 2013.
8. Brief-history-of-machine-learning https://www.dataversity.net/a-brief-history-of-machine-learning/#
9. Mahesh, Batta. (2019). Machine Learning Algorithms -A Review. 10.21275/ART20203995.
10. **Shreyas D K1, Srivatsa N Joshi2, Vishwas H Kumar3, Vishaka Venkataramanan4,**." A Review on Neural Networks and its Applications”: journal of computer technology &application. SSN:2229-6964 (Online).ISSN: 2347-7229 (Print).Volume 14, Issue 2, 2023.DOI (Journal): 10.37591/JoCTA
11. Dr.C K Gomathy, Kakamanu Jaya SairamMacmillan, Illuru Yadhu Vamsi,” A THE HANDWRITTEN DIGITS RECOGNITION”. International Journal of Engineering Applied Sciences and Tocology, 2021.Vol. 6, Issue 7, ISSN No. 2455-2143, Pages 203-206.Published Online November 2021 in IJEAST
12. Хайкин С. Нейронные сети. Полный курс. Вильямс, 2018, 1104 с.
13. Murphy Kevin P. Machine Learning: A Probabilistic Perspective. The MIT Press, 2012,1104c.
14. TensorFlow. Информационный портал.[Электронный ресурс]. Режим доступа: https://www.tensorflow.org/
15. TensorFlow.Guide Обучающие материалы [Электронный ресурс]. Режим доступа: https://www.tensorflow.org/guide/keras. Дата обращения: 28 февраля 2020.
16. Fadhil Abbas Fadhil, Mohamed Ali, Nikolai Safiullin, “ The study on usage of table functions instead of basic operators inside encryption algorithm “ , 2022 Ural-Siberian Conference on Biomedical Engineering, Radio electronics and Information Technology (USBEREIT), Yekaterinburg, Russian Federation, 2022, pp. 320-323, doi:10.1109/USBEREIT56278.2022.9923412.
17. Syed Sohail Ahmed, Zahid Mehmood, Imran Ahmad Awan, Rehan Mehmood Yousaf, "A Novel Technique for Handwritten Digit Recognition Using Deep Learning", Journal of Sensors, vol. 2023, Article ID 2753941, 15 pages, 2023. https://doi.org/10.1155/2023/2753941.
18. Keras datasets https://storage.googleapis.com/tensorflow/tf-keras-datasets/mnist.npz.
19. Goodfellow, Ian J., Bulatov, Yaroslav, Ibarz, Julian, Arnoud, Sacha, and Vinay Shet. "Multi-digit Number Recognition from Street View Imagery using Deep Convolutional Neural Networks." ArXiv, (2013).
20. Dahl, George E., et al. "Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition." IEEE Transactions on audio, speech, and language processing 20.1 (2011): 30-42.