Handwritten equation solver using Convolutional Neural Network

Chapter · September 2023

DOI: 10.1201/9781003453406-6

Pavinder Yadav
National Institute of Technology, Hamirpur

Nidhi Gupta
National Institute of Technology, Hamirpur

Handwritten Equation Solver using Convolutional Neural Network

Mitali Arya[a], Pavinder Yadav[a], Nidhi Gupta[b,*]

[a]National Institute of Technology Hamirpur, 177005, Himachal Pradesh, India.

[b]National Institute of Technology Kurukshetra, 136119, Haryana, India.

[*]Corresponding Author - *nidhi.gupta@nitkkr.ac.in*

Abstract

Because of the ongoing COVID-19 pandemic, educational activities resulted in an unanticipated change from traditional learning to a digital teaching and learning environment. E-learning is the delivery of educational content and learning via digital resources. It increases the digitization of handwritten documents because students are required to submit their homework and assignments online. This work proposes an automatic system for handwritten numeric recognition and equation solver based on Convolutional Neural Network to assist teachers and parents in checking handwritten assignments. Handwritten digit recognition refers to the ability of computer machine to recognise handwritten digits from various sources such as published researchs, real-world images, touch display, and so on. In this work, the CNN model is used to recognize and solve handwritten equations that contains four basic arithmetic operations, addition, subtraction, division and multiplication. The handwritten linear equations with some limitations are also being solved by the proposed model.

Keywords

Handwritten Equations, Deep Learning, CNN, Automation, Image Processing

1 Introduction

It is a difficult task in image processing to use a Convolutional Neural Network (CNN) to create a robust handwritten equation solver. Handwritten mathematical expression recognition is one of the most difficult problems in the domain of computer vision and machine learning. In the field of computer vision, several alternative methods of object recognition and character recognition are offered. These techniques are used in many different areas, such as traffic monitoring [3], self-driving cars [9], weapon detection [17], natural language processing [11], and many more.

Deep learning is subset of machine learning in which neural networks are used to extract increasingly complex features from datasets. The deep learning architecture is based on data understanding at multiple feature layers. Further, CNN is another core application of deep learning approach, consisting of convolutions, activation functions, pooling, densely linked, and classification layers. Over the past several years, deep learning has emerged as a dominating force in the field of computer vision. When compared to classical image analysis problems, CNNs have achieved the most impressive outcomes.

Deep learning is becoming increasingly important in today's era. Deep learning techniques are now being used several fields like handwriting recognition, robotics, artificial intelligence, image processing, and many others. Creating such a system necessitates feeding our machine data in order to extract features to understand the data and make the possible predictions. The correction rate of symbol segmentation and recognition cannot meet its actual requirements due to the two-dimensional nesting assembly and variable sizes. The primary task for mathematical expression recognition is to segment and then classify the characters. The goal of this research is to use CNN model that can recognise handwritten digits, characters, and mathematical operators from an image and then set up the mathematical expression and compute the linear equation.

The purpose of this study lies in designing a deep learning model capable of automatical recognising handwritten numerals, characters, and mathematical operations when presented with an image of the handwriting. In addition, the purpose extends to built a calculator that is capable of both setting up the mathematical statement and computing the linear equation.

The article is divided into different sections. Section 2 presents thorough summary of current handwritten character recognition research studies in recent years. Section 3 goes through each component of the CNN in depth. Section 4 describes the proposed deep learning algorithms for handwritten equation recognition as well as the dataset used. Section 5 discusses the comparative analysis of different technical approaches. In addition, future scope and conclusion of the work are provided in Section 6.

2 State-of-the-Art

There are variety of methods that have been developed to recognise handwritten digits. Handwritten digit recognition has many applications such as bank cheques, postal mails, education etc. Many methods have been used to recognize handwritten digits such as Support Vector Machines (SVM), Naive Bayes, CNN, K-Nearest Neighbors etc. in the past years. In a few decades, CNN has achieved good performance in handwritten digit recognition.

For offline handwritten character recognition (HCR), Agarwal *et al.* [4] employed CNN and Tensorflow. They divided HCR system into six stages: image data collection, image preprocessing for enhancement, image segmentation, feature extraction using the CNN model, classification, and postprocessing fo detection. Furthermore, Softmax Regression was used to assign probabilities to handwritten characters because it produces values ranging from 0 to 1 and sums to 1. The use of normalization in conjunction with feature extraction resulted in higher accuracy results as achieved more than 90%. However, the study did not provide any information on the specific comparative outcomes and the dataset that was examined.

Bharadwaj *et al.* [6] used Deep Convolution Neural Networks for effective handwritten digit detection on Modified National Institute of Standards and Technology(MNIST) dataset. The dataset is consist of 250 distinct forms of writing and 70,000 digits. The proposed technique includes the steps of preprocessing, model construction and compilation, training the model, evaluation of trained model, and detection of digits. Both the computer-generated and handwritten digits were recognized by the model.They predicted real-world handwritten digits with 98.51% accuracy and 0.1% loss. However, the model could only identify the characters from the clear and good quality images. The most difficult component is dealing with the images that are blurred or have noise in the real-world images.

Thangamariappan *et al.* [16] used a variety of machine learning techniques for handwritten digit recognition, including Naive Bayes, Random Forest, SVM and others. The model was trained using a multi-layer perceptron neural network model on the MNIST dataset. They achieved 98.50% accuracy in Digit Recognition with MNIST dataset. Meanwhile, the test accuracy on the same dataset was 88.30%, which was relatively low when compared to the training accuracy.

Chen *et al.* [14] used CNN to recognise four basic arithmetic operations which are addition, division, subtraction, and multiplication. On the MNIST dataset, the CNN's performance was trained and evaluated. The improved CNN model is tested in handwritten digit recognition and four arithmetic operations. The convergence speed of CNN model has been reduced and observed 91.20% accuracy. The authors only experimented on clear images in their proposed model, which was trained on the MNIST dataset. The trained model was unable to recognise characters in noisy or blurry images.

Gawas *et al.* [8] proposed a system for recognising handwritten digits and symbols that consists of addition, subtraction, and multiplication and solving basic equations containing these operations. They used CNN and created a front-end interface that allows the user to write the equation, which is then identified and solved. They used libraries such as OpenCV, Keras, and Flask to deploy the model. The trained model could only perform fundamental mathematical operations but failed to solve linear equations.

The authors created their own dataset for the handwritten equation solver and trained it using deep learning models. By constructing the equation solver calculator, researchers not only identifies the characters and symbols, but also has solved the linear mathematical problem.

3 Convolutional Neural Network

A Convolutional Neural Network (CNN), in certain cases referred to as ConvNet, is a type of deep neural network that specializes in image processing and has a grid-like topology [5]. A CNN is a feed-forward neural network with multiple layers. It is formed by assembling many layers on top of one another in the sequence which can be seen in Fig. 1. CNN trains the model using raw pixel image input, then extracts features for better categorization.

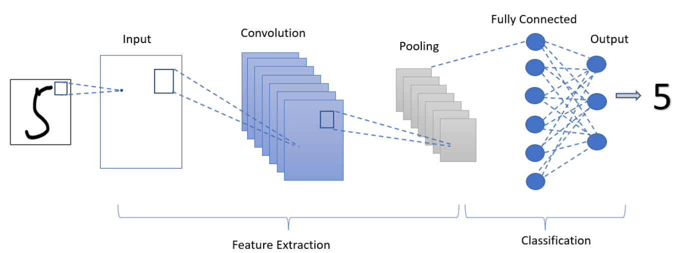


Figure 1: CNN architecture for handwritten images.

Convolutional layers, fully connected layers, and pooling layers make up the three different kinds of layers that are included in a deep neural network.

3.1 Convolution Layer

The first layer utilised to extract various information from input images is the Convolutional Layer. The dot product is computed using an array of data input and a two-dimensional array of weighted parameters known as a kernel or filter, as shown in Fig. 2.

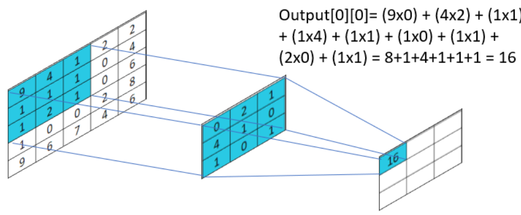


Figure 2: Convolution layer.

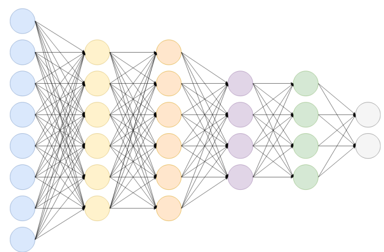


Figure 4: Fully connected layer.

3.2 Pooling layer

This layer is generally used to make the feature maps smaller. It reduces the number of training parameters, which speeds up computation. There are mainly three kinds of pooling layers: Max Pooling- It chooses the maximum input feature from the feature map region as shown in Fig. 3, Average Pooling- It chooses the average input feature from the feature map region, and Global Pooling- This is identical for employing a filter with the dimensions h x w, i.e., the feature map dimensions.

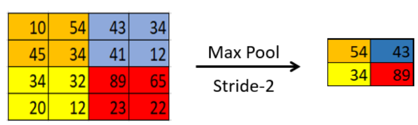


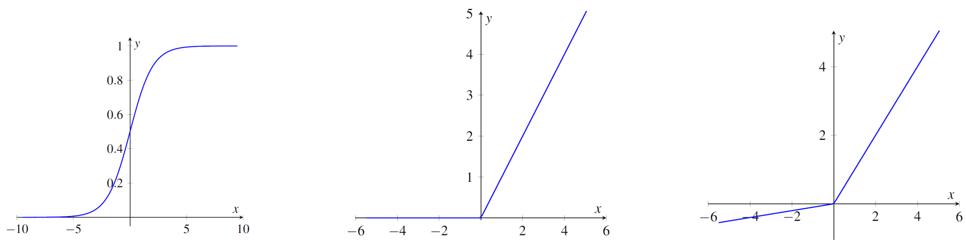
Figure 3: Max pooling.

3.3 Fully Connected Layer

The last few layers of the neural network are Fully Connected Layers. As shown in Fig. 4, if the preceding layer is entirely linked, every neuron in that layer is coupled to every other neuron in the layer below it. In our proposed method, two fully connected layers in CNN are employed followed by the classification layer.

3.4 Activation Function

In simple words, Activation Function, shown in Fig. 5, activates the neurons. It helps in deciding whether or not a neuron should fire and determining the output of the convolution layer. These are the most common activation functions: Sigmoid [12], ReLU [13], Leaky ReLU [7], and Softmax [10].



(a) Sigmoid (b) ReLU (c) LeakyRelu

Figure 5: Several different kinds of activation functions

The sigmoid and SoftMax activation functions are represented in Equations 1 and 2, respectively.

$$\sigma(x) = \frac{1}{1+e^{-x}} \tag{1}$$

$$\operatorname{Softmax}(x_i) = \frac{\exp(x_i)}{\sum_j \exp(x_j)} \tag{2}$$

4 Handwritten Equation Recognition

4.1 Dataset Preparation

The first and most important step of any research is dataset acquisition. The numerals and operations data and the character/variable dataset was collected from Kaggle [1, 2]. Then it was augmented to prepare large dataset. The dataset contains approximately 24,000 images which has 16 classes, like 0-9 numerals, variable and five basic mathematical operators/symbols, namely, addition, subtraction, multiplication, equals, and division as shown in Fig. 6.

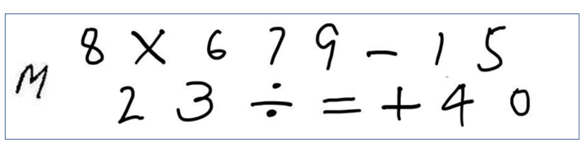


Figure 6: Sample images in the dataset

4.2 Proposed Methodology

The proposed CNN model is used to recognize simple equations which consists of arithmetic operators that are addition, subtraction, multiplication and division. It is also used to recognize simple linear equations of the type $x + a = b$ where x is a variable and a, b are constants. The block diagram of implemented model is illustrated by Fig. 7.

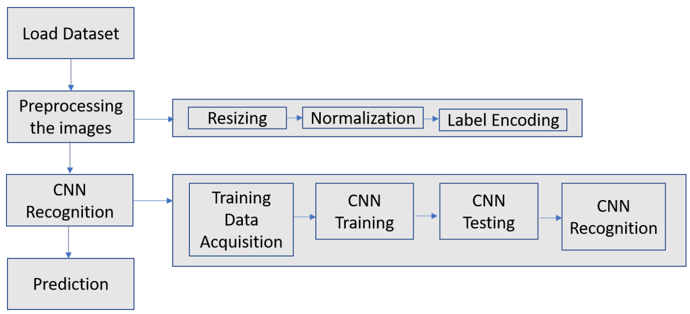


Figure 7: Block diagram of proposed scheme for handwritten digits

4.2.1 Dataset Acquisition

The dataset contains approximately 24,000 handwritten images divided into two sets- training images and testing images. Number of training images is approximately 1,300 whereas testing images are taken to be approximately 50 for each category. Pre-processing which includes resizing, cropping of images, padding etc. was done to make the dataset uniform. The images have a resolution of 95 x 84 for digits and 94 x 89 for the character M. The images were further resized to 100 x 100 for smooth training and better results.

4.2.2 Preprocessing

The goal of preprocessing is to improve the image quality to analyse it more effectively. Image are preprocessed using several well-known techniques, like image resizing, normalizing and augmentation are a few examples.

(i) Image Augmentation- It is a technique of artificially increasing dataset size. Image augmentation use various techniques of processing or a combination of multiple methods of processing, such as random rotation, shear, shifts, flips, and so on.

(ii) Image Resizing- In CNN, model accepts all the images of the same sizes only. Therefore, all images need to be resized to the fixed size. The images have been resized to 100 x 100 for sequential model and 224 x 224 for inception model.

(iii) Normalization- Normalization is a process that alters the intensity range of pixels. The primary goal of normalisation is to make computational task more efficient by reducing values from 0 to 1.

(iv) Label Encoding- It is the process of translating labels into a numeric representation that machines can read.

4.2.3 Recognition through CNN Model

Handwritten datasets were used for training data acquisition after being supplemented with various methods such as shearing, rotating, nearest filling, shifting, and so on. There are approximately 23,000 training sample images and 950 testing sample images in the handwritten numeral dataset. To increase the variability and diversity of training data, we deformed it using several transformations like rotation, translation, scaling, vertical and horizontal stretching. By adding a few samples to make the dataset more diverse, the goal is to make it easier for CNN to find handwritten digits when the dataset is used to train CNN.

4.2.4 Processing inside CNN Model

The model was trained using sequential approach. The sequential model builds the structure layer by layer. The model contains seven Conv2D layers, four MaxPooling2D layers, six drop-out layers with the rate 0.2 (Fraction of the input units to drop).

In addition, the activation function of the employed convolution layer was modified from Sigmoid to ReLU Activation function, and subsequently to Leaky ReLU Activation function. Leaky ReLU speeds up training and it has other benefits over ReLU too. As it is a multi-class classification issue, the Softmax Activation function was utilised in the final dense layer. The model was then optimised using the Adam optimizer.

Also, the model is trained using inception architecture [15]. InceptionV3 is a convolutional neural network-based deep learning image categorization algorithm. The input layer, 1x1 convolution layers, 3x3 convolution layers, 5x5 convolution layers, max pooling layers, and combining layers are the parts that make up an inception model. The module is simple to unpack and comprehend when broken down into the constituent parts.

4.3 Solution Approach

The solution approach is shown using flowchart in Fig. 8. The handwritten mathematical equation is being provided by user.

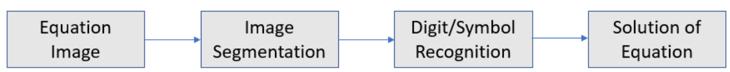


Figure 8: Sequence of proposed solution approach

Image segmentation is the process of dividing an image information into divisions known as image segments, which helps to minimise the computational complexity and makes the further processing or analysis easier. The segmentation stage of an image analysis system is crucial because it isolates the subjects of interest for subsequent processing such as classification or detection. Image categorization is used in the application to better accurately classify image pixels. Fig. 9 represents the actual deployment of the proposed methodology. The input image is segmented into well-defined fixed proportions. In the case of simple character recognition provided in the image, we have segmented it into three parts, i.e., into two numerals and one operator. This case is considered in general form of image. The segmentation is done into 1:3 ratio. The proposed model works with the constraints having the middle segment should be an operator and the extreme segments should belong to numerals.

The Fig. 10 describes the steps in the algorithm of the proposed model to solve the equations from handwritten images. In both cases, each of the segment is thresholded by Otsu's Algorithm. Later, the segmented binary image is normalized before fed to the model for training. The size of segmented image is further defined by four coordinates, as left, right, top, bottom. Each segment will be now framed into new image named as segs. Each of these segments are resized into 100x100.

Now, the segmented character/variables or operators are extracted and recognized by the trained model. The end goal of the training is to be able to recognize each block after analyzing an image. It must be able to assign a class to the image. Therefore, after recognizing the characters or operators from each image segment, the equation is being solved using mathematical formulas on trained model.

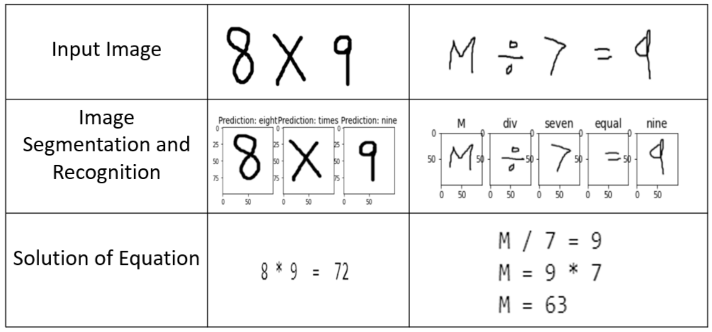


Figure 9: Practical implementation of proposed scheme

Algorithm 1 Algorithm for Handwritten Equation Solver using CNN Model

Input: Image of Handwritten Linear Equations
Image Cropping: Crop the input image into the required segments; n is number of segments.
for $i = 1$ *to* n **do**
    height, width = size(img)
    div = int(width / n)
    left = i * div
    top = 0
    right = (i+1) * div
    bottom = height segs = img.crop(left, top, right, bottom)
Image Resizing: Resized to 100 X 100 for uniform training set size
cropped_segs = segs.resize(100,100)
Thresholding:
bin_segs = cv2.threshold(cropped_segs)
Normalization: By dividing all pixel values by the highest pixel value
bin_segs = bin_segs / 255
Output: Each blocks are recognized seperately containing characters or operators.

Figure 10: Algorithm for handwitten equation solver.

5 Results and Discussion

The results of the experiments show that CNN can correctly segment handwritten typefaces and then combine the results into an equation. It is capable of recognising fundamental operations. Instead of manually recognising handwritten digits and symbols, the trained CNN model can recognise basic four arithmetic operations, digits, and characters effectively. In terms of checking mathematical equations, the CNN model has a relatively stable performance. The model was trained through both Sequential Approach and Inception Architecture. The comprehensive outcomes of both models are shown in Table 1 below.

The performance of the proposed model is observed not efficient for 10 epochs for model loss and validation loss. Therefore, both models are further trained for 30 epochs to observe the accurate results.

Table 1: Comparison between sequential model and inception model

Model	training Accuracy	training Loss	Validation Accuracy	Validation Loss
Sequential Model(10 epochs)	98.10%	5.98%	74.30%	129.63%
Sequential Model(30 epochs)	99.46%	1.66%	99.20%	3.38%
Inception Model(10 epochs)	97.32%	9.38%	98.59%	6.39%
Inception Model(30 epochs)	99.42%	1.83%	99.50%	2.70 %

However, both the model gave similar results in case of 30 epochs. The graph for Sequential's model for 30 epochs is shown in Fig. 11. The model accuracy is 99.46% with 1.66% model loss.

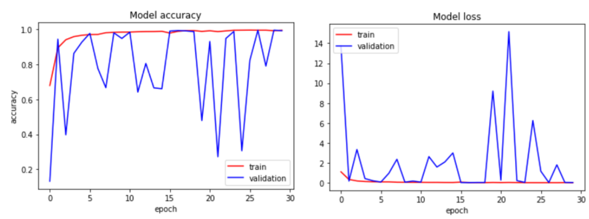


Figure 11: Accuracy and loss of sequential model

The graph for Inception's model for 30 epochs is shown in Fig. 12. The accuracy of model is observed 99.42% with 1.83% model loss.

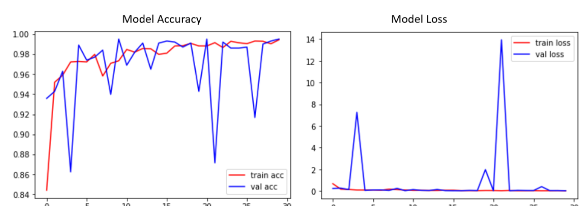


Figure 12: Accuracy and loss of inception model

The proposed model functions correctly on handwritten equations, regardless the handwriting style, i.e., whether, good or bad. Even if the equation is written in messy handwriting; the proposed model is able to detect it correctly as shown in Fig. 13. In this example of poor handwriting, we can see digit '8' is not written in good handwriting, even though model is detecting it accurately and able to solve the equation. However, despite of good efficiency, the proposed model posses some limitations as well.

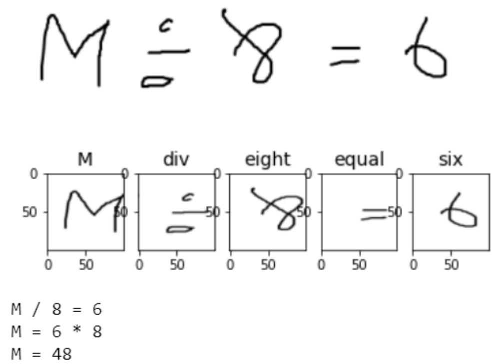


Figure 13: Sample of poor handwritten equation

6 Conclusion and Future Scope

Handwritten Digit Recognition and Equation Solver have been implemented using Convolutional Neural Networks. By replacing activation functions of the CNN architerure algorithm with a Leaky ReLU, an improved CNN algorithm is proposed through Sequential Approach. Both Sequential Model and Inception model have been used for experimentation to observe the results. Both the models produced better results for 30 epochs rather for 10 epochs. The proposed CNN model with Leaky ReLU is tested for handwritten numeral recognition, where it is used to automatically check four basic arithmetic operations, addition, subtraction, multiplication, and division. Till now, it has been trained to solve handwritten simple linear equations on a single character/variable only.

Also, the handwritten equation solver must be capable of recognising the characters and operators from the input images as quickly as possible. Hence, there is a need of Graphics Processing Unit (GPU) for the large dataset to significantly reduce the training as well as testing time. The overall recognition accuracy of the CNN based handwritten recognition model is observed as 99.46%. The future work may be extended to solve handwritten quadratic equations from the images. Also, the work may include to solve equations having more than one character/variable.

References

[1] Dataset. https://www.kaggle.com/code/rohankurdekar/handwritten-basic-math-equation-solver/data.

[2] Dataset. https://www.kaggle.com/datasets/vaibhao/handwritten-characters.

[3] Mahmoud Abbasi, Amin Shahraki, and Amir Taherkordi. Deep learning for network traffic monitoring and analysis (ntma): A survey. *Computer Communications*, 170:19–41, 2021.

[4] Megha Agarwal, Vinam Tomar Shalika, and Priyanka Gupta. Handwritten character recognition using neural network and tensor flow. *International Journal of Innovative Technology and Exploring Engineering (IJITEE)*, 8(6S4):1445–1448, 2019.

[5] Saad Albawi, Tareq Abed Mohammed, and Saad Al-Zawi. Understanding of a convolutional neural network. In *2017 international conference on engineering and technology (ICET)*, pages 1–6. Ieee, 2017.

[6] Yellapragada SS Bharadwaj, P Rajaram, VP Sriram, S Sudhakar, and Kolla Bhanu Prakash. Effective handwritten digit recognition using deep convolution neural network. *International Journal of Advanced Trends in Computer Science and Engineering*, 9(2):1335–1339, 2020.

[7] Arun Kumar Dubey and Vanita Jain. Comparative study of convolution neural network's relu and leaky-relu activation functions. In *Applications of Computing, Automation and Wireless Systems in Electrical Engineering*, pages 873–880. Springer, 2019.

[8] Jitesh Gawas, Jesika Jogi, Shrusthi Desai, and Dilip Dalgade. Handwritten equations solver using cnn. *International Journal for Research in Applied Science and Engineering Technology (IJRASET)*, 9:534–538, 2021.

[9] Abhishek Gupta, Alagan Anpalagan, Ling Guan, and Ahmed Shaharyar Khwaja. Deep learning for object detection and scene perception in self-driving cars: Survey, challenges, and open issues. *Array*, 10:100057, 2021.

[10] Ioannis Kouretas and Vassilis Paliouras. Simplified hardware implementation of the softmax activation function. In *2019 8th international conference on modern circuits and systems technologies (MOCAST)*, pages 1–4. IEEE, 2019.

[11] Daniel W Otter, Julian R Medina, and Jugal K Kalita. A survey of the usages of deep learning for natural language processing. *IEEE transactions on neural networks and learning systems*, 32(2):604–624, 2020.

[12] Andrinandrasana David Rasamoelina, Fouzia Adjailia, and Peter Sinčák. A review of activation function for artificial neural network. In *2020 IEEE 18th World Symposium on Applied Machine Intelligence and Informatics (SAMI)*, pages 281–286. IEEE, 2020.

[13] Johannes Schmidt-Hieber. Nonparametric regression using deep neural networks with relu activation function. *The Annals of Statistics*, 48(4):1875–1897, 2020.

[14] Chen ShanWei, Shir LiWang, Ng Theam Foo, and Dzati Athiar Ramli. A cnn based handwritten numeral recognition model for four arithmetic operations. *Procedia Computer Science*, 192:4416–4424, 2021.

[15] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.

[16] P Thangamariappan and JC Pamila. Handwritten recognition by using machine learning approach. *Int. J. Eng. Appl. Sci. Technol*, pages 564–567, 2020.

[17] Pavinder Yadav, Nidhi Gupta, and Pawan Kumar Sharma. A comprehensive study towards high-level approaches for weapon detection using classical machine learning and deep learning methods. *Expert Systems with Applications*, page 118698, 2022.