International Journal on
"Technical and Physical Problems of Engineering"
(IJTPE)
Published by International Organization of IOTPE

ISSN 2077-3528
IJTPE Journal
www.iotpe.com
ijtpe@iotpe.com

December 2023 | Issue 57 | Volume 15 | Number 4 | Pages 245-255

STUDYING THE PERFORMANCE OF CONVOLUTIONAL NEURAL NETWORKS IN RECOGNIZING PRINTED WORDS

H.A.A. Alsaadi M. Cevik

Department of Electrical and Computer Engineering, Altinbas University, Istanbul, Turkey
203720034@ogr.altinbas.edu.tr, Mesut.cevik@altinbas.edu.tr

Abstract- This article gives a brief introduction to the area of document evaluation and recognition, outlining the main steps and resources that go into creating recognition systems. It also gives a broad introduction to the subject area. We concentrate on the basic operations of such a system, delving into the specifics of a single character identification challenge in addition to describing a variety of different ways in which convolutional neural networks might be put to use. In addition, we are conducting a formal comparison between two widely used methods for creating neural networks of the next generation that are capable of recognizing printed characters, specifically feed forward backpropagation neural Kohonen Self-Organizing Features Map (KSOFM) and Multilayer Perceptron (MP).

Keywords: OCR, CNN, Deep Learning, Performance Analysis, Printed Words Recognition.

1. INTRODUCTION

Convolutional neural networks (CNN), together with their theoretical underpinnings, learning algorithms, architectural intricacies, and real-world applications, have attracted a lot of interest during the past ten years. Excellent reading material for newcomers may be found in both [1] and [2]. Convolutional Neural Networks are utilized in a variety of document evaluation and activity recognition at various stages throughout the recognition and analysis processes. The growing number of paper documents used in administrative processes across all levels of business, from companies and government departments to educational colleges and private households, has generated a significant amount of interest in optical character recognition system applications. These systems are used to digitize handwritten text from paper documents. Because computers can store, process, searching, as well as extract knowledge from vast amounts of electronic data, systems which are capable of the mechanical conversion of written text into electronic form appear to be of significant relevance. Because of their ability to yield relatively low classification errors, neural networks have found widespread application in this field. This is especially true when they are applied to data that is both disturbed and noisy.

This is one of the reasons why neural networks have become so popular in this field. Recent developments in the field in question have necessitated the creation of specialized approaches to render CNN usable there. In a system like this one, the optical characters recognition module is necessary because it is the one in charge of recognizing every individual character. The human brain processes information while reading texts in blocks of letters in a specific context to use some prior knowledge about the subject area. A recent research study demonstrates that using additional context, even characters or words that are difficult to make out in noisy text may be detected fast. These procedures are extremely important to the operation of document analysis systems. However, the rate of classification of the most fundamental system module, which can classify a single character, ought to be optimized to the greatest extent possible. Because of this, research into the algorithms used for recognizing written text is extremely pertinent.

Comparison of the capabilities of self-organizing features maps (KSOFM) with multilayer feed-forward perceptions has importance in both a practical and theoretical context (MP). Inside the scope of this research, we investigate the architectures to analyze, contrast, and evaluate how well they recognize patterns included within images of written letters. Abstract features such as the relative location of components in character or connection patterns as well as their count were not included in our tests because we supplied the raster picture into the KSOFM and MP [2] networks as a one-dimensional array. This meant that our tests did not include those abstract features. It was shown that MP was more effective at solving problems involving character identification when the input data was explained.

An overview of each section of this paper can be found in the following outline: This article provides a synopsis of the documents analytics plus recognition system in a short format. We focused on the application of neural networks in the various processes involved in the processing of document images. The apparatus that was utilized in the tests to evaluate the functionality of neural networks is described, and the results of those evaluations are displayed once the studies have been completed. The paper should include both a discussion section and a concluding paragraph.

2. STUDY AND RECOGNITION OF DOCUMENTS

The difficulty with research methodologies and recognition is complicated by the presence of a great deal of subproblems. According to [3], we are able to identify some of the fundamental components of such a system as well as the function that each of these components serves. During the preprocessing phase of document images, several picture adjustments such as noise reduction, brightness balance, and skew detection and correction are carried out. In this multi-stage approach, the "physical layout analysis" entails searching for groups of information that are comparable to one another (often text or pictures, but also mixed types and special types like signature). The areas of the map that are determined to be compatible with a certain recognition type are tagged so that they can be processed by the appropriate module. An analysis of the logical layout is the name given to this stage of the procedure.

The information about the detected region is subsequently sent by the system to one of three factors: word recognition, signature separation, or graphic item segmentation. After the signature has been segmented, the resultant bounded picture is submitted for validation to the module that is responsible for signature recognition. The final step in the process of digitizing an image is the graphic item recognition module, which receives its input from the graphic segmentation module. After words have been broken down into their component parts, they can then be recognized in a variety of methods, such as by the detection and segmentation of individual characters, through word recognition, or through a methodology that integrates data on previously recognized words and characters. Figure 1 illustrates both the individual components of the system as well as its overall operation.

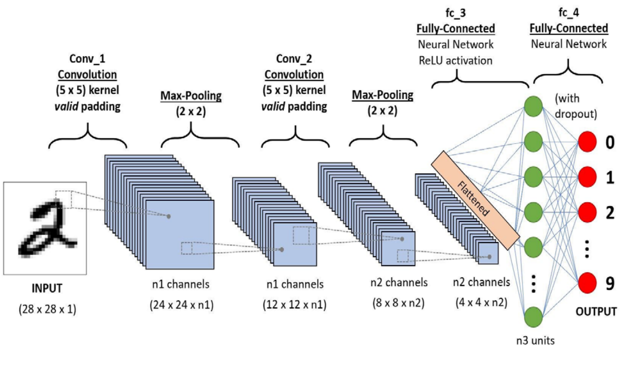

Figure 1. Detecting documents. CNN workflow steps are ovals, diamonds represent staged information [3]

2.1. Design Evaluation and Urban Design

In contrast to the situations that typically call for the application of Convolutional Neural Networks, which typically involve the following categories, segmentation techniques are typically broken down into the following subcategories: page classification, pixel categorization, and area identification. The success rate of picture segmentation is drastically impacted by the prevalence of skewed images. The image that is provided into the zone identification module should not have any unwelcome qualities, such as merged or broken graphics or characters, or a changing background. Because of this, finding these traits is crucial for effective picture analysis [4]. Virtual MP networks are effectively used to remedy the skew issue. A network is then trained to output the skew angle based on the extracted attributes of the image. A de-skew can be accomplished at this angle. The quality of textual images can also be enhanced with text restoration. Repairing a merged or otherwise corrupted text is the job of the "text restoration" step. This is a crucial first step that will aid in subsequent document analysis stages. Kalman filtering [5], morphological filtering [6], and Line following [7] are common methods for restoring textual images. Though it has been said that MP networks can be put to use for filtering, this is still something that needs further investigation. MP network is given pixels from a fixed size moving window to achieve this. Figure 2 depicts the filtering procedure. Since a single trained neural network cannot handle the wide variety of document pictures, it is usual practice to retrain an MP network for each page individually.

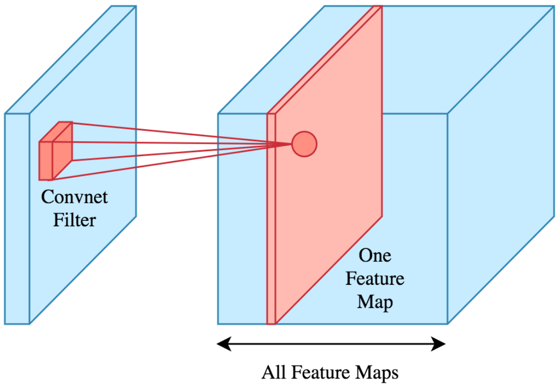

Figure 2. Neural network-based filtering to obtain cleared pixel values on the output image [7]

Pixel categorization can be broken down into its most fundamental form, which involves assigning a background or foreground name to each individual pixel [8]. Utilizing pixel classification eventually made it possible to classify text, graphics, and lines apart from one another. Neural networks have been utilized by a number of authors [9, 10] for the purpose of pixel categorization. The extraction of the region's interest points and the subsequent use of those data as inputs into a linear classifier are the standard steps involved in regional classification [11]. When attempting to solve this problem for a particular region, neural networks make use of both local and global features. Calculations are made for every pixel to determine the regional features of the area [12]. RBF networks have been shown to perform better than MP networks, KSOFM networks, and stochastic neural networks when it comes to this task [13].

The categorization of websites is a difficult problem with a wide variety of potential answers. For earlier approaches, it was essential to have a pre-printed form layout that included ruling lines [14]. When it comes to document types in the business world, the typical suspects include business letters and technical papers; however, in recent years, the categorization of book and journal pages has managed to garner considerable interest [15]. Business letters and technical reports are the usual culprits when it comes to document types in the business world. Depending on the zones that are present, you may either use a tree [16] or a graph [17] to describe the layout of a page. When training a neural network for classifications, it is customary to use recurrent neural networks and to create a feature representation of a fixed size in order to get around the restriction of a small input window. This is done so that the problem won't arise in the first place.

2.2. Analyzing each Character in Detail

Signature segmentation is required for the verification process to proceed. Without the need for a combination of dynamism [18] and structure [19], it is difficult to avoid skilled forging. Alterations to the line's grayscale throughout its length and the width of the signature's stroke are both examples of dynamic factors. As a result, such disparities must be gleaned in order for the verification process to be successful. Most signature verification systems, as a direct consequence, center their attention on the local and global unifying principles features of signature structures.

2.3. Performing Operations on Text and Images

The problem of assigning a specific class to a region in an image or text can be treated as a classification challenge involving two classes. Whether or not a specific document region should be treated as text or treated as non-text can be determined in a number of different ways. There are primarily two methods that one could approach fixing this problem: from the very highest level or from the very lowest one. Top-down approaches frequently make use of projection profiles [20, 21] or run-length smoothing [22, 25]. Note: An image is disassembled into pieces, which are then recognized and converted in to the text column, sentences, text lines, and, ultimately, words [23]. These steps are performed working from the top of the image down. When applied to text that is not formatted in a perfect square, this method does not produce accurate results. Bottom-up techniques often involve iteratively grouping components beginning at the pixel level and working their way up [26, 27]. The following level of structure is formed when these clusters develop into the words, lines, and paragraphs that make up the following level [28]. Neural networks might be helpful in solving problems like these [29].

2.4. Analysis of Words' Meanings and Recognizing their Forms

The word recognition approach deals with the recognition of whole words without segmentation; the character segmentation and recognition approach divide the image into regions that match the properties of individual characters; and the integrated approach is a combination of the word recognition approach and the character segmentation and recognition approach; it integrates the recognition of specific characters with the recognition of whole words. The process of character segmentation is one area that could stand to profit from the implementation of neural networks (locating the cutting places and identifying the touching characters). The outcomes of our trials are pertinent at this juncture since we are evaluating the character recognition capabilities of a variety of networks, and our findings will help us do so.

3. EXPERIMENTAL CONDITIONS AND OUTCOME

In order to make an accurate comparison between both the KSOFM and the MP network, we scanned in a sheet that had 16 machine-printed lines, and on each of those lines had the full Alphabetic characters written in upper case letters. The page was black and white. This allowed us to make an accurate comparison between the two systems. Each of the character lines contains 26 characters, and each one of those characters was printed using a different typeface than the others. The technique that was used to classify the text was based on the presumption that there is none "active" pixel to pixel that really are suitably dark between rows of text in separate texts lines or between letters within the same line of text. This was the foundation for the strategy that was used. This was the concept that the plan was built upon, thus it was very important. The use of antialiasing allowed for the individual characters to be scaled up or down. During the tests, a scaling method was used so that the characters could be placed inside a box with the same height and width on all sides. After antialiasing, the brightness of some pixels decreases; for example, the black pixels that are located at the letter edges become too brilliant and thin to give the same neural network with sufficient additional information. To circumvent this issue, nonlinear contrast correction is used immediately after the resizing procedure has been finished.

Because of this procedure, grayscale pixels will appear to be black. This set of adjustments results in a larger text size with a reduced number of visible artifacts, much like an algorithm that restores characters. In order to evaluate the abilities of neural networks and examine the differences between the two architectures, we designed and carried out a series of experiments in which the parameters were varied. One of the key issues was the level of precision at which a neural network could recognize characters with which it was not previously familiar. When creating the images for our characters, Designers used squares with pixel dimensions of 14×14, 16×16, and 18×18 respectively. The representation of the input view will ultimately define the dimension of something like the neural network's processing system. In the second inquiry, it was requested that an investigation into how the accurate recognition rate is affected by the size of a particular data collection, such as the train data set, be carried out. The first five and first eight fonts shown in Figure 3 made up the data sets that we used.

1	ABCDEFGHIJKLMNOPQRSTUVWXYZ
2	ABCDEFGHIJKLMNOPQRSTUVWXYZ
3	ABCDEFGHIJKLMNOPQRSTUVWXYZ
4	ABCDEFGHIJKLMNOPQRSTUVWXYZ
5	ABCDEFGHIJKLMNOPQRSTUVWXYZ
6	ABCDEFGHIJKLMNOPQRSTUVWXYZ
7	ABCDEFGHIJKLMNOPQRSTUVWXYZ
8	ABCDEFGHIJKLMNOPQRSTUVWXYZ
9	*ABCDEFGHIJKLMNOPQRSTUVWXYZ*
10	ABCDEFGHIJKLMNOPQRSTUVWXYZ
11	ABCDEFGHIJKLMNOPQRSTUVWXYZ
12	ABCDEFGHIJKLMNOPQRSTUVWXYZ

Figure 3. Characters set: Trained neural networks employed two typeface sets [38]

We were able to identify the ideal collection of eight and five font styles, including both, that provide the best potential learning for the network by having trained able to receive nets with various configurations of fonts supplied into the network in different sequences. As a result, we were able to train receptive networks with various typefaces input into the network in various sequences. Because of this, we were able to determine the level of learning that the network was capable of. We aren't interested in discovering the optimal combination of fixed size fonts for showing full data sets; instead, we desire to know how much the majority of the data set impacts the recognition accuracy. This is because we want to display whole data sets. Specifically, we want to know how the recognition rate is influenced by the size of the data set. These were the same experimental variables that were used in both the KSOFM and the MP neural network studies.

The third characteristic that distinguished MP networks from other types was the depth of a hidden layer. We chose three sizes: fifty neurons, seventy-five neurons, and one hundred neurons. The fact that an MP with any given hidden layers can really be simplified to a network that has just one convolution layers is known knowledge, we implemented this reduction using a single hidden layer in our approach. Because an MP can be simplified to a network, this action was taken. In the end, we decided on 1500 and 3000 as the optimal values for the maximum number of learning epochs for each MP-specific factor. As a consequence of this, there will be a total of 36 individual trials for the MP network. Due to the fact that the third parameter defines the sizes of the Kohonen layers, to 30 and 60 pieces, correspondingly., we were able to gather a total of 12 distinctive testing from the KSOFM network. In order to draw any conclusions that are even remotely significant, a minimum of 20 separate trials of each experiment were carried out.

When it comes down to the nitty-gritty details, the Encog library was the one who carried out the experiments [30]. The tanh protocol was utilized in the activation of our MP network's single hidden layer. The learning rate and the momentum were both maintained at a level of 0.3 throughout. Neighborhood The local functionality of Bubble was leveraged to provide power to the KSOFM network. The neighborhood value for the function that was supplied was decided to be set at the value 5. We developed a linear reduction by deciding the learning rate would be 0.01 and the neighborhood radius would be 1. Because there was a consistent amount of network error by the 10th or 20th epoch, further training was deemed pointless after that point, and the count of epochs was increased to 50. Due to the fact that KSOFM networks use unsupervised learning, it is difficult to employ them effectively in the role of classifier. We chose the path of least resistance.

After the training phase has been completed, the network is questioned with all of the information that it has acquired. Keep a record of the neurons for each letter that did the best matching operations while going through that process. By applying the filter, get rid of BMUs that have won for more than just one letter. After that, the network ought to go through some testing. In the event that the network generates a BMU that is not known to the persistent ones, locate the BMU that is the closest to the newly picked neuron using Euclidean distance. In this part of the analysis, we make use of the clustering capabilities of the KSOFM network by making the assumption that neurons that are geographically close to one another (relative to the metric system) are utilizing the same character categorization.

Table 1 shows the outcomes of a KSOFM tests, while Table 2 shows the outcomes of the MP network tests. When it comes to recognition, MP networks do exceptionally well, as evidenced by the fact that their recognition rate grows linearly with the number of epochs. This is shown by the recognition performance.

When we discuss the recognition rate of a font, we are referring to the proportion of characters across all fonts that were successfully detected. This is the total number of fonts that were evaluated. The size of the amount of data in both tables that depicts the frequency at which the recognition was identified (the bottom row in Table 2 and the bottom two rows in Table 1) which illustrates how well the outcomes fared in relation to previous studies' findings. The number is denoted as having the highest recognition rate when it is underlined in bold, while the recognition rate with the lowest rate is given when the number is underlined with a small letter size. As the number of fonts used for training grows, researchers find that subjects achieve higher recognition scores while simultaneously seeing a reduction in the number of buried neurons. On the other hand, if you use a training set with only 5 fonts and an input image size of 18×18 pixels, you will achieve the worst possible performance.

The KSOFM results are shown in Table 2, which demonstrates that the recognition rate decreases as the overall size of the source image decreases. The larger the input image, the higher the quality of the outputs that can be produced by the network. According to the results of the KSOFM tests, there appears to be an intriguing relationship between the identification rate and the layer size of Kohonen neurons. Increases in the amount of a network's layers are proportional to increases in the rate at which it recognizes devices. In a similar vein, it does not make a difference how large the image that is being received is. Both linkages have occurred at the appropriate period due to the self-cauterizing qualities possessed by the KSOFM networks. When mapping higher-dimensional input neurons to the lower-dimensional space of a Kohonen network, there is greater room for creative expression than in other mapping methods.

Table 1. Test fonts mean recognition rates in (%) for multilayered perception [30]

MP		Size of the Images																	
		14×14						16×16						18×18					
		Characters Count up for Training						Characters Count up for Training						Characters Count up for Training					
		Neurons hidden			Neurons hidden			Neurons hidden			Neurons hidden			Neurons hidden			Neurons hidden		
		50	75	100	50	75	100	50	75	100	50	75	100	50	75	100	50	75	100
Detection Rate %	1500 Training Cycles	59	54	49	43	38	27	67	59	54	47	41	28	73	68	62	50	47	35
	3000 Training Cycles	61	59	52	43	40	34	67	63	56	50	42	39	75	69	67	53	48	43

Table 2. Test fonts mean recognition rates in (%) self-organizing features map [30]

KSOFM	Size of the Images											
	14×14				16×16				18×18			
	Characters Count up for Training				Characters Count up for Training				Characters Count up for Training			
	5		8		5		8		5		8	
	Neuronal Output		Neuronal Output		Neuronal Output		Neuronal Output		Neuronal Output		Neuronal Output	
	30	60	30	60	30	60	30	60	30	60	30	60
Recognition Rate %	15	37	18	17	13	30	21	21	17	18	19	20

3.1. Proposal System

The proposed solution adapts to each individual user based on CNN's methodology. CNNs are a subfield of deep learning that is frequently employed for this purpose. It is a simulation designed from the ground up to imitate the mind's operation. A typical CNN has three layers: one for input, multiple for processing in the background, and one for output. The hidden layer structure consists of convolutional layers, function activation layers, layers for pooling, fully linked layers, and normalization layers. CNN requires less preprocessing than competing classifiers and improves with additional training. Figure 1 depicts the CNN sampling network's general layout. In a fully linked ANN, the processing is performed in layers, commencing with those used for feature extraction and concluding with those used for classification. Numerous layers are utilized to obtain a high level of recognition accuracy, which is resistant to even minor geometric changes in the input images. Due to its high recognition accuracy, CNN has been effectively utilized for classification of real-world data.

The method was designed for use in a specialized educational app that aids educators in teaching numerical concepts. The goal of this method is to accurately identify the digit the user has written. The user is tasked with writing the answer to a question in one area of the app. Therefore, each student's handwriting samples will be gathered as input before beginning to learn a digit. For practice with the number 2, the program can have the pupil write it down multiple times. This information is fed into the custom-built CNN with the students' handwriting when they use the app to learn the specified digit. The student's initial application writing serves as a sample for instruction, while subsequent application writing serves as a test sample. In this research, we focus on training based on the individual. That means we need a CNN dedicated to each individual issue. There should be n10 different CNN architectures if there are nth topics. This is because there are 10 numbers (0-9) to check. Figure 4 shows 10 (0-9) CNN built from the same subject's data.

3.2. Preparation

More information on the architecture's structure is provided here. The photographs are first subjected to several pre-processing techniques such as resizing, cropping, etc. Once the essential preparations have been made, the data can be input into the system.

This argument will be further developed in the part concerning the dataset. Each of the layers regarding the convolutional neural network (CNN) alters the volume of activations across a differentiable function. We used a total of fifteen levels and seven different layers to build the network. A layer for classification, a layer for pooling, a layer for ReLU, a layer for SoftMax, a layer for convolution, and a layer for batch normalization are present. Figure 5 shows the general layout of the system and the final placements of the layers.

Because of its duty as a feature extractor, the feature of that specific Neural Network (NN) is formed in the very first block. This is accomplished by employing convolution filtering techniques for template matching. In the first layer, images are filtered using convolution kernels, which results in "feature maps" that are either normalized using an activation function or scaled. We must first submit data to the system. Image Input Layers are used to enter two-dimensional images into a network and normalize the data that results. When scaled to its 'zero center' position, its dimensions are 100 by 186 by 1. The submitted image has a range of 100 to 186 in terms of resolution, with a value of 1 denoting grayscale. Then, the input images are passed via a convolution layer, which is a stack of convolutional filters. Each one of these filters emphasizes a different quality of the input photographs.

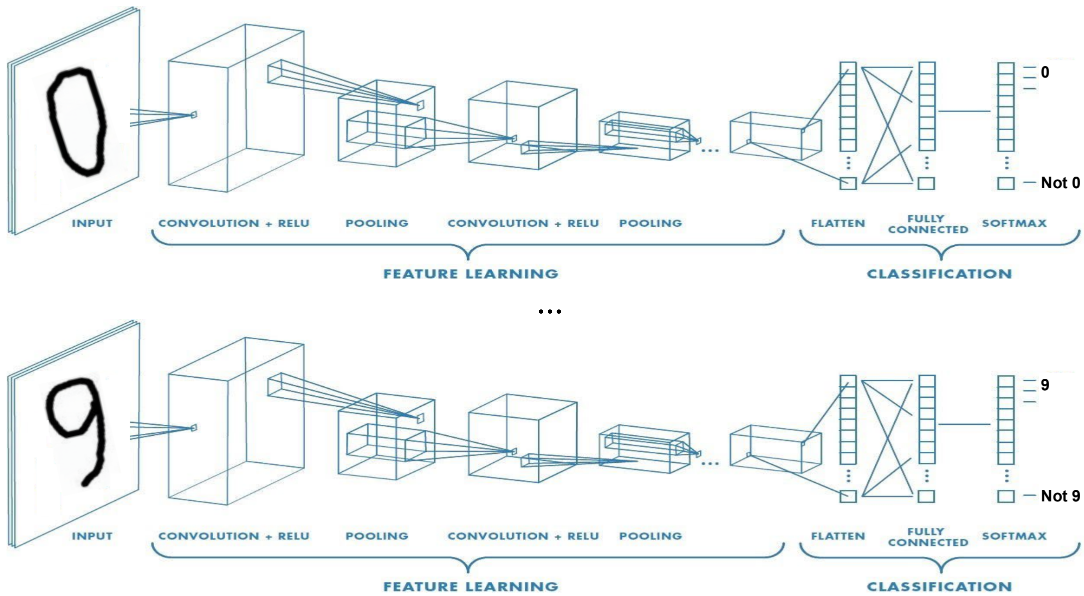

Figure 4. 10 subjects specific CNNs comprise the system architecture

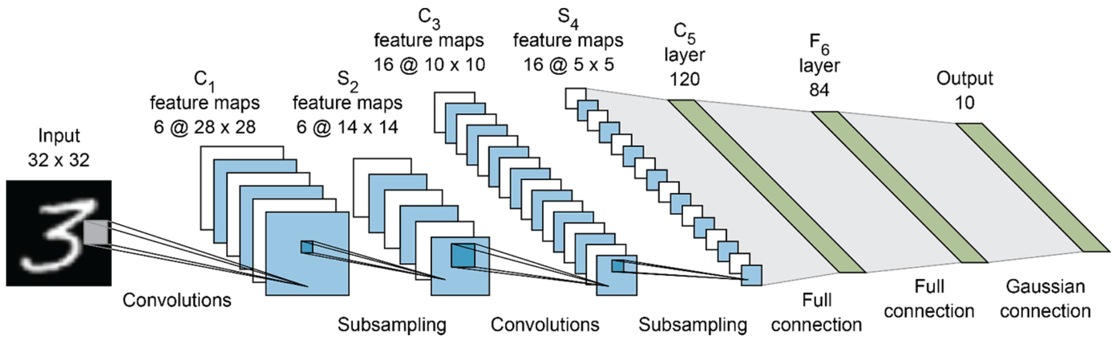

Figure 5. Experiment with the CNN model that was proposed

There are eight filters in this scenario, and each filter is three by three. Convolutions for padding [0 0 0 0] and stride [1 1] are also supported. After the convolution layer, the normalization layer for the batch is applied to normalize all the input channels within a small batch.

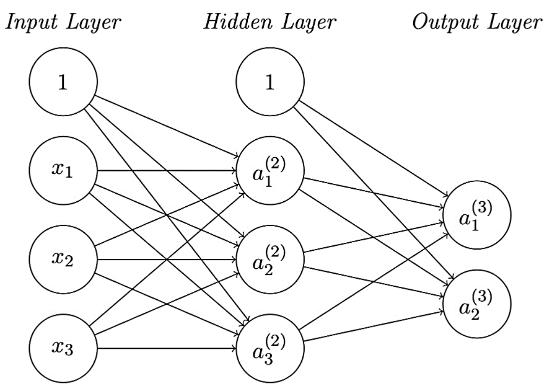

Figure 6. ReLU layer performs threshold operations

This, when put between the convolutional and ReLU layers, speeds up CNN training and decreases network initialization vulnerability. The activation layer, or fourth layer, is referred to as a ReLU Layer. As shown in Figure 6, a ReLU layer performs threshold operations on each input element, setting any value less than zero to zero.

The input is divided into rectangular pooling sections, and the maximum for each zone is identified, followed by the addition of a layer called maximum pooling together with pools sized [2 2] as stride [2 2]. The second unit is a layer repeat that does not include the input layer. The convolution layer is the only distinguishing feature in this situation. It has parameters that differ from the original. A 2-dimensional convolutional layer is built using 32 filters of size [3 3] and the 'same' padding. During training, the software decides the size of each of the zero paddings, making the layer output proportionate to the input.

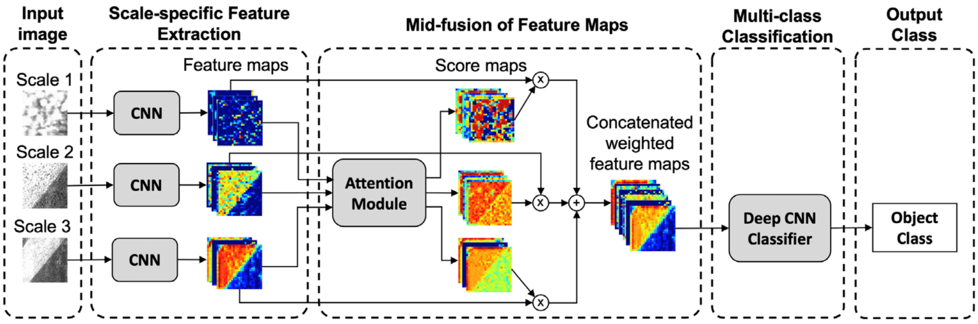

Figure 7. Our CNN setup's design for a single-digit target

The third and final block's convolution layer makes use of both the batch Standardization Layer and the RELU Layer from the subsequent block. As a result of the RELU Layer's multiplication of the input data by the weights matrix and the addition of the bias vectors, the network produces a fully connected layer. The final two layers are output layers. Because we're dealing with a verification problem, our model can only have two states: true or false. For problems relating to classification with independent classes, the classification layer which calculates the cross-entropy loss is the top layer. Figure 7 depicts the architecture of our CNN system for a single participant and a single number.

People who have to rely on special education services tend to have illegible handwriting. Three students with minor mental retardation provided the handwriting samples shown in Figure 8. It's clear that every kid has a unique style in their handwriting. In addition, many numerical values are extremely difficult to anticipate when presented in a non-sequential format. For instance, in the first topic, 7 is like 4, and in the second, 6 is like 0. Scan the third topic with a standard HWR system, and you'll see a 0 that looks like a 9 written backwards. The subjects also showed a general lack of competence in adhering to standard conventions for writing numerals. For instance, in response to the instruction to write the number 8 without lifting a hand, the third subject interlaced the two rings to create the 8. None of the third topic's samples have the expected writing direction. If we examine the third subject's attempt to write the number 9 without raising their hand, we can see that he or she did it by drawing an anchor on the perimeter of the circle. Traditional HWR algorithms fail miserably when presented with handwritten examples from pupils at this level for the reasons given above. Successful results have been achieved, however, because our method is person-dependent and generates a dataset from one's handwriting.

The test computer's GPU is low-end, and the system is intended for use in an instructional program for tablets, therefore the image resolution and other settings had to be adjusted so that the app could be run rapidly. Therefore, it is believed that the computer will do better if tested on a higher powerful machine. The accuracy also depends on the size of the training set and improves as more data is used. In this case, we employed a small dataset for both training and testing purposes. The less the effect of training error as well as test error, and the greater overall accuracy, more data should be included in training set.

Figure 8. Handwritten examples from three students' personalities

4. CASE STUDY

1. Prepared Images: Input images from any real-world application, including those containing handwritten characters, are contaminated with noise. Image pre-processing, including noise removal, is a crucial component in optical character recognition. Noise reduction is followed by image segmentation. The size and shape of the cropped segmented image will be adjusted as necessary. It is necessary to change color images to black and white occasionally.
2. Segmenting Images: By segmenting the input image into sections, we are able to more easily extract features such as objects and textures. The process of separating apart distinct parts of a picture is called "segmentation." Image segmentation is commonly used for anomaly detection in healthcare as well as remote sensing. Image segmentation makes use of a wide variety of techniques, including line, point, edge, threshold, area-based, and pixel-based clustering. Researchers face a number of obstacles when attempting to use picture segmentation with a convolutional neural network.
3. Extracting Features: The number of variables in huge data sets is expected to be substantial. It will require a great deal of processing power. To facilitate data administration, it is suggested that the level of detail of the initial incoming raw data be reduced. Feature extraction refers to the process of selecting and/or merging variables into features without compromising system efficiency or dataset completeness.
4. Classification: Is the process of giving names to things that haven't been observed yet. This is achieved through

training a deep learning algorithm on a data collection containing relevant samples. A supervised learning issue, classification involves associating a label with each sample. It is often written as an integer that cannot be negative.

5. Dataset: When training a CNN model, the MNIST dataset of handwritten characters is employed. The MNIST dataset consists of 70,000 pictures, all with a resolution of 28 by 28 pixels. The English alphabet is visually represented by 23,342 capital and 22,928 lower case characters. The dataset utilized to evaluate is a collection of handwriting samples from a variety of people.

The following are the stages of handwritten English character recognition:

1) Bring in the handwritten English alphabet pictures from the MNIST database.
2) The downloaded dataset should be divided into training and test images.
3) The MNIST dataset undergoes some preliminary image processing, including a greyscale conversion.
4) The dataset should be normalized.
5) Separate the data set into subsets of varying sizes, each of which will be used for training.
6) The pool window size, filter size, and number of convolution and pooling layers will all be experimented with when the labelled data is utilized to train a CNN model.
7) The best trained model is used to classify handwritten English characters.
8) Different epochs of model execution will be investigated for their effects on training and validation accuracy.

Modifying elements like the filter size, learning rates, number of filters, number of convolutional and pooling layers, and pool window size all aid in accelerating the creation of the training model. For validation, the model with the best results was chosen. The CNN model with four convolutional layers, four max pulling layers, a four-by-four convolution filters, category crossover entropy loss, and the Adam optimizer produced the best results after a total of fifty iterations. The epoch-specific results for the accuracy of training and validation are displayed in Table 3. Training and validation reliability are shown to rapidly improve between epochs 10 and 50. Even when using more than 50 epochs, the accuracy numbers are essentially unchanged.

Table 3. Accuracy in Training and Validation

	Existing CNN Model with three convolution layers, three maximum pooling layers, and a four-by-four convolution filter		Proposed CNN model with four convolution layers, four maximum pooling layers, and a four-by-four convolution filter	
EPOCHS	Percentage of Accuracy in Training	Percentage of Validation Accuracy	Percentage of Accuracy in Training	Percentage of Validation Accuracy
10	88.45	89.78	92.83	93.44
20	90.25	90.78	96.12	95.73
30	91.44	90.56	97.36	96.38
40	92.38	90.63	98.16	96.38
50	93.49	92.88	98.65	96.54

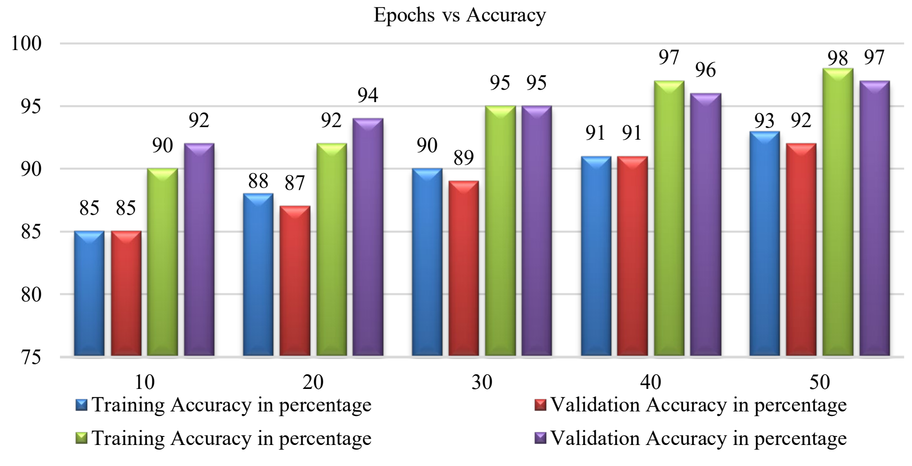

Figure 9. Epochs vs Accuracy

A crucial first step in processing data for many real-world applications is recognizing handwritten entries. The use of convolutional neural networks has substantially benefited in the recognition of handwritten English characters. The system can recognize English characters more accurately than earlier systems thanks to the addition of layers made possible by pooling and convolution, as well as the adjustment of numerous hyperparameters. Layers of various types are shown in Figure 9 and are commonly employed in CNNs.

1. The input is protected by the input layer before being passed on to the convolutional layer below it.
2. Data features are extracted thanks to the convolutional layer. Like a traditional neural network, a convolution is

an ordered procedure that multiplies the data input by any number of weights. Due to the size difference between the filter and the input data, we must use the dot product to multiply what's inside a filter-sized patch of the information that was entered. Until a single number remains, dot products are computed by repeatedly multiplying items from the input with filter pieces of the same size. Since the result of this operation is always a single scalar value, the term "scalar product" is commonly used to describe it. Filters smaller than the list of inputs can be used to effectively apply the same filtering (a collection of variables) to multiple points throughout the input array. Each overlapped section or filtering-sized patching of the input data receives a linear application of the filter from left to right and top to bottom.

3. CNN architecture has a plethora of activation functions. To improve the efficiency of the learning model, the activation function known as Rectified Linear Unit (ReLU) is used to replace the negative value with zero.

4. The Pooling layer compresses each feature map to lessen the network's computational load.

5. A "fully connected", a layer in which all the neurons in the layer can send and receive messages with all the neurons in the layers below it. Only a layer with complete connectivity can interpret the non-linear interaction of the attributes. For categorizing written English characters, we use the results from our fully linked layer.

6. CONCLUSIONS

The consequences of the experiments conducted on both networks point to a connection between the recognition rate and the operating parameters of the network. When it comes to MP networks, lower-resolution input images perform better, but when it comes to KSOFM networks, the opposite is true. The fundamental nature of networked systems is the most important factor in explaining this phenomenon [35]. It is necessary to provide the KSOFM network with an input image that has a high resolution for it to function at its optimal level when performing classification. When the size of the input field is allowed to increase unchecked from a particular starting position, it is reasonable to anticipate a decline in recognizability. In a similar vein, if you reduce the dimension of the input layer towards the conclusion of the process without first training the network, it will be unable to recognize letters effectively. The fact that we are working with some of the most crucial optical data lies at the center of almost all of these problems. In addition, in contrast to the Cognitron and Noncognition neural networks [36], our neural networks do not utilize receptive fields in any way. These networks are able to take in visual data as a result of the ease with which abstract features, such as joints and their orientation, may be retrieved using these networks. In OCR systems, this is often handled by a preprocessing module that uses traditional KSOFM and MP networks to extract such abstract notions from character images. These networks are used to recognize the characters. Therefore, there are two possible directions for future research and development:

1) Improvements in neural designs that resemble the human sense of sight in terms of data hierarchy processing (by extracting abstract features).

2) The incorporation of data preprocessing modules that can extract abstract ideas despite differences in character size, location, or alignment, and afterwards feed this information into common modern classification algorithms like MP or KSOFM.

There is potential for major progress to be made along both growth paths. Both possible future courses are very clear to us right now. Take into consideration that in many documents analysis methodologies, KSOFM and MP are envisioned with one network's output serving as the inputs to the next. This is something that must be always kept in mind. Even though Cognition and Noncognition demand a greater amount of processing power than other methods (which is a concern for mobile computing), they are also able to replace a significant number of simpler connected components. As the development of such systems continues, we might reach a stage where we need only a few (1-3) modules to store complex hierarchical networks. If we get to that point, it will be feasible for us to do so. These two potential courses of action could be investigated in greater depth as potential solutions to the problem of OCR.

REFERENCES

[1] A.A. Abdulhakeem, O.K. Raafat, "The Impact of IOT on Real-World Decisions in the next Stage", AIP Conference Proceedings, Issue 1, Vol. 2591, pp. (020029)1-10, 2023.

[2] C. Krittanawong, H. Zhang, Z. Wang, M. Aydar, T. Kitai, "Artificial Intelligence in Precision Cardiovascular Medicine", Journal of the American College of Cardiology, Vol. 69, Issue 21, pp. 2657-2664, 2017.

[3] L.T. Le, H. Nguyen, J. Dou, J. Zhou, "A Comparative Study of PSO-ANN, GA-ANN, ICA-ANN, and ABC-ANN in Estimating the Heating Load of Buildings' Energy Efficiency for Smart City Planning", Applied Sciences, Vol. 9, Issue 13, p. 2630, 2019.

[4] A.A. Abdulhakeem, "Improve the Performance of the CNPV Protocol in VANET Networks", International Journal of Civil Engineering and Technology (IJCIET), Vol. 9, Issue 11, pp. 304-314, 2018.

[5] B. Jabir, N. Falih, "Big Data Analytics Opportunities and Challenges for the Smart Enterprise", International Journal on Technical and Physical Problems of Engineering (IJTPE), Issue 47, Vol. 13, No. 2, pp. 20-26, June 2021.

[6] B.S. Aziz, A.A. Jassim, "Modeling of Industrial Catalytic Naphtha Reforming Plant by Using Artificial/ Neural Network (ANN)", Journal of Environmental Science, Computer Science and Engineering and Technology, Vol. 3. No. 4, pp. 2170-2185, India, 2014.

[7] Kh. Bellat, A. Mansouri, S. Raihani, "Implementation of Artificial Neural Network for Optimization of a Wind Farm", International Journal on Technical and Physical Problems of Engineering (IJTPE), Issue 47, Vol. 13, No. 2, pp. 35-39, June 2021.
[8] D. Manamalli, P. Kanagasabapathy, K. Dhivya, "Expert Optimal Control of Catalytic Reforming Using ANN", Journal of Chem. Eng. Comm., Vol. 193, pp. 729-742, 2006.
[9] D. Scherer, A. Muller, S. Behnke, "Evaluation of Pooling Operations in Convolutional Architectures for Object Recognition", International Conference on Artificial Neural Networks, Springer, pp. 92-101, 2010.
[10] K. Ahmed K. Aljabery, M.A. Hussein, A.A. Abdulhakeem, "Dynamic Routing for Multi-Hop Wireless Network", International Journal of Computer Science and Mobile Applications (IJCSMA), Vol. 10, Issue 8, pp. 80-92, 2022.
[11] E.H. Hssayni, M. Ettaouil, "A Novel Pooling Method for Regularization of Deep Neural Networks", International Conference on Intelligent Systems and Computer Vision (ISCV), pp. 1-6, 2020.
[12] E.H. Hssayni, M. Ettaouil, "New Approach to Face Recognition Using Co-Occurrence Matrix and Bayesian Neural Networks", The IEEE 6th International Conference on Optimization and Applications (ICOA), pp. 1-5, 2020.
[13] E.H. Hssayni, M. Ettaouil, "Regularization of Deep Neural Networks with Average Pooling Dropout", Journal of Advanced Research in Dynamical and Control Systems, Vol. 12, No. 04-sp, pp. 1720-1726, 2020.
[14] G. Zahedi, S.M. Zadeh, G. Moradi, "Enhancing Gasoline Production in an Industrial Catalytic Reforming Unit Using Artificial Neural Networks", Journal of Energy and Fuels, Vol. 22, No. 4, p. 2671, 2008.
[15] H. Xiao, K. Rasul, R. Vollgraf, "Fashion-MNIST: A Novel Image Dataset for Benchmarking Machine Learning Algorithms", arXiv Preprint, arXiv: 1708.07747, 2017.
[16] K.M. Liang, H.Y. Guo, S.W. Pan, "A Study on Naphtha Catalytic Reforming Reactor Simulation and Analysis", Journal of Zhejiang, Vol. 6, No. 6, pp. 590-596, 2005.
[17] A. Krizhevsky, I. Sutskever, G.E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks", Advances in Neural Information Processing Systems, pp. 1097-1105, 2012.
[18] L.S. Carvalho, C.L. Pieck, M.S. Rangel, N.S. Figoli, J.M. Grau, et al., "Trimetallic Naphtha Reforming Catalysts; Properties of the Metal Function and Influence of the Order of Addition of the Metal Precursors on Pt-Re- Sn/[gamma]-Al2O3- Cl", Appl Catal, pp. 91-103, 2004.
[19] M. Ranzato, Y.L. Boureau, Y. LeCun, "Sparse Feature Learning for Deep Belief Networks", Advances in Neural Information Processing Systems, pp. 1185-1192, 2008.
[20] M. Sun, Z. Song, X. Jiang, J. Pan, Y. Pang, "Learning Pooling for Convolutional Neural Network", Neurocomputing, Vol. 224, pp. 96-104, 2017.
[21] M. Zile, "Improved Control of Transformer Centers Using Artificial Neural Networks", International Journal on Technical and Physical Problems of Engineering (IJTPE), Issue 40, Vol. 11, No. 3, pp. 28-33, September 2019.
[22] M.D. Zeiler, R. Fergus, "Stochastic Pooling for Regularization of Deep Convolutional Neural Networks", arXiv Preprint, arXiv: 1301.3557, 2013.
[23] M.N. Raheema, "Multilayer Neural Network Learning by Genetic Algorithm", M.Sc. Thesis, Department of Electrical and Electronic Engineering, University of Technology, Baghdad, Iraq, August 2000.
[24] M.A. Hussein., E.K. Lubna, A.A. Abdulhakeem, "Optimum Placement of Nodes in Networks of Wireless Sensors", AIP Conference Proceedings, Issue 1, Vol. 2591, pp. (020019)1-10, 2023.
[25] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting", The Journal of Machine Learning Research, Vol. 15, No. 1, pp. 1929-1958, 2014.
[26] P. Sermanet, S. Chintala, Y. LeCun, "Convolutional Neural Networks Applied to House Numbers Digit Classification", The IEEE 21st International Conference on Pattern Recognition (ICPR2012), pp. 3288-3291, 2012.
[27] R. Beaulah Jeyavathana, "Deep Learning Applications and Challenges", International Journal of Control and Automation, Vol. 13, No. 2s, pp. 79-83, 2020.
[28] G. Li, E. Geng, Z. Ye, et al., "Indoor Positioning Algorithms are Based on the Improved RSSI Distance Model", Sensors, Vol. 18, Issue 9, pp. 20-28, 2018.
[29] S. Abdufattokhov, K. Ibragimova, M. Khaydarova, A. Abdurakhmanov, "Data-Driven Finite Horizon Control Based on Gaussian Processes and its Application to Building Climate Control", International Journal on Technical and Physical Problems of Engineering (IJTPE), Issue 47, Vol. 13, No. 2, pp. 40-47, June 2021.
[30] S. Ioffe, C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift", arXiv Preprint, arXiv: 1502.03167, 2015.

BIOGRAPHIES

Name: **Hussain**
Middle Name: **Abd Ali**
Surname: **Alsaadi**
Birthday: 01.09.1996
Birthplace: Karbala, Iraq
Bachelor: Computer Engineer, Al Hussain University College, Karbala, Iraq, 2017
Master: Student, Electrical and Computer Engineering Department, Faculty of Engineering, Altinbas University, Istanbul, Turkey, 2022
Research Interests: Machine Learning, AI
Scientific Memberships: Iraqi Engineers Syndicate

Name: **Mesut**
Surname: **Cevik**
Birthday: 06.10.1970
Birthplace: Sarkisla, Turkey
Bachelor: Electronics Engineer, Electronics Engineering Department, Faculty of Engineering, Istanbul University, Istanbul, Turkey, 1993
Master-1: Electronics Engineer, Electronics Engineering Department, Graduate School of Natural and Applied Sciences, Istanbul University, Istanbul, Turkey, 1996
Master-2: Electrical Engineer, Electrical Engineering Department, School of Engineering and Applied Sciences, Columbia University, NYC, USA, 2001
Doctorate: Electrical and Electronics Engineer, Electrical and Electronics Engineering Department, Graduate School of Natural and Applied Sciences, Istanbul University, Istanbul, Turkey, 2007
The Last Scientific Position: Assist. Prof., Electrical and Electronics Department, Engineering and Architecture Faculty, Altinbas University, Istanbul Turkey, Since 2021
Research Interests: Wireless Communications, AI, Machine Learning, Deep Learning
Scientific Publications: 25 Papers, 3 Chapter Books
Scientific Memberships: IEEE