Dissertation Submitted for the partial fulfillment of the **B.Sc. as a part of M.Sc. (Integrated) Five Years Program AIML/Data Science** degree to the Department of AIML & Data Science.

Project Dissertation

ANALYZING AND SOLVING HANDWRITTEN MATHEMATICAL EQUATIONS USING AI

submitted to

By

ZEEL RATHI
Semester-VI

M.Sc. (Integrated) Five Years Program AIML/Data Science

Department of AIML & Data Science.
School of Emerging Science and Technology
Gujarat University

June 8, 2022

DECLARATION

This is to certify that the research work reported in this dissertation entitled "<u>ANALYZING AND SOLVING HANDWRITTEN MATHEMATICAL EQUATIONS USING AI</u>" for the partial fulfilment of B.Sc. as a part of M.Sc. (Integrated) in Artificial Intelligence and Machine Learning degree is the result of investigation done by myself.

ACKNOWLEGEMENT

On this occasion of submit my research work, I would like to thank all the people who have made this possible. The role of Department of Applied Mathematical Science, Actuarial Science & Analytics, Gujarat University in shaping my professional capability is very important.

I thank Dr. Ravi Gor, co-coordinator of the Department of Applied Mathematical Science, Actuarial Science & Analytics, for all his support. He has always supported me and encouraged me to overcome my difficulties. I am extremely grateful to him for his valuable suggestions and guidance he has provided me during my research work.

Rashmi Ma'am has always guided me on the right path and has been a motivating source for innovative research work. This work would have never been possible without her guidance, support, and encouragement. With the help of her guidance, I magnificently overcame the difficulties during my work and learned at each step.

I am heartly and lovingly thankful to my parents for their sacrifice, support and endless love which encouraged me to efficiently overcome the difficulties in my pursuit of the research and who always helped me in my work.

Last but not the least, I would not be able to complete the work without the blessings of almighty and would like to thank God for giving me the strength and patience to carry out my work with full of dedication and ethical values.

~ Zeel Rathi

INDEX

CHAPTER 1

ABSTRACT

Using CNN to create a robust handwritten equation solver is a difficult task in image processing. One of the most difficult challenges in computer vision research is handwritten mathematical expression recognition. The work is made more difficult by the fact that certain characters are segmented and classified. Convolutional Neural Networks Characters are classified using a network. Each appropriate is required for the solution of the problem. Character string operation is used for detection. The recognition of HMSE becomes not only an ambitious task but a motivating research area covering concepts of computer vision, pattern recognition, feature extraction, and artificial intelligence. Finally, the results of the experiment show that the strategy we've described is quite effective.

Keywords: Mathematical expressions, handwritten mathematical symbols and expressions, Handwriting recognition, Classification techniques, Machine learning, Deep learning

CHAPTER 2

INTRODUCTION

Due to the swift improvement in computer technology and internet technology, most of the documents, books and literatures in the area of computer science as well as others are increasingly becoming digitalized. Mathematics is broadly used in almost all areas of science, such as physics, engineering, medicine, economics, etc. Digital document analysis and understanding is the major research concern today. For the recognition of English characters and numbers in electronic books OCR (optical character recognition) can attain higher recognition exactness. Handwritten mathematical expression recognition is still a most challenging job to do in the area of computer vision. Due to the two-dimensional nesting assembly and different sizes, the correction rate of symbol segmentation and recognition still cannot achieve its actual requirements. The primary task for the recognition of mathematical expression is to segment the character and then classify those character. An operation that seeks to decompose an image of a sequence of characters into sub images of individual symbols is Character segmentation.

Convolutional neural network (CNN) is one of the mostly used classification model in computer vision area. In the last few years, deep Convolutional Neural Network (CNN) leaning has proved the outstanding performance in the field of image classification, machine learning and pattern recognition. Above all existing model, CNN is one of the most popular models and has been providing the state-of-the-art recognition accuracy on object recognition, segmentation, human activity analysis, image super resolution, object detection, scene understanding, tracking, and image captioning.

CHAPTER 3

OBJECTIVE

The objective is to perform an extensive state of the art on the techniques and methods used for recognizing and classifying HMSE. The authors endeavour to bring out all significant findings in sub-processes, representation models, algorithms, tools, datasets, and comparative analysis of the accuracy of the recognition models.

1. The system should be able to process the mathematical expression in advance.

2. The system should be able to recognize text in the image as well as mathematical symbols.

3. The system should extract text from the image and display the mathematical expression's solution.

4. The primary goal is to learn how to create a CNN model for Mathematical Expression.

CHAPTER 4

RELATED WORK

1) Handwriting Recognition

Our project spans the field of image classification. By comparing the effectiveness of each method of learning multi-class discriminative models, we link together years of image classification research.

Bernard et. al proposed the Random Forest classifier. We extend this research on the MNIST dataset to our own MNIST, HaSy, and Kaggle composite dataset.

Zhu et al. extended the standard AdaBoost binary classification problem to a multi-class classification problem. We combine this with the work of Schapire et al. in our two-stage AdaBoost classifier.

One of the best works in handwritten digit recognition comes from LeCunn et al in his seminal work on hand written character recognition from Springenberg et al. attempt.

Springenberg et al. attempts to simplify and generalize CNN architecture by replacing max pooling layers with more convolutional layers. Clevert et al. proposes the Exponential Linear Unit and compares it to its counterparts. We also apply the research of batch normalization by Ioffe et al. and dropout by Srivastava et al.to optimize and accelerate training our deeper networks without over-fitting.

Significantly less work has been done on handwritten mathematical expression recognition. Prior to CROHME, the relatively small number of math recognition research was done without benchmark data sets, standard encodings, or evaluation tools; this made progress slow and collaboration difficult for the community. The CROHME competition, organized since 2011, seeks to make it much easier to get started working on handwritten math recognition and 1 meaningfully compare systems.

2) Convolutional Neural Networks Our approach relies heavily on CNNs, which are widely used for a variety of vision recognition problems. Many papers document ways of achieving better results when training/evaluating CNNs, including using slant correction on images, elastic distortions to increase the size of the training data set and robustness of the model, and extracting additional features from images as inputs to the CNN. We will use some of these methods in our approach to the problem.

CHAPTER 5

DATASET

The 2013 Competition on Recognition of Online Handwritten Mathematical Expressions (CROHME) was won by Vision Objects (now named My Script), with 60% accuracy at the expression level; second place only achieved 23% accuracy. Vision Objects is a privately held company with proprietary technology, so while public-domain research has the potential to reach much higher accuracies, it is currently unclear how this can be achieved. The poor results in CROHME reflect the general state of handwritten expression recognition public research. Further, to the best of our knowledge, no papers have been published applying Convolutional Neural Networks (CNNs) to the task of handwritten expression recognition.

Our project investigates the problem of recognizing handwritten mathematical expressions, which we also chose for our CS221 final project. Our primary contribution is in creating an end-to-end system using a well-trained CNN model to go from strokes to symbols to a LATEX expression.

OUR MODEL DATASET DESCRIPTION

1) Dataset consists of jpg files (28 x 28).
2) For simplicity, we are using 0–9 digits, +, — and, times images in our equation solver.
3) On observing our dataset, we can see that it is biased for some of the digits/symbols, as it contains 12000 images for some symbol and 3000 images for others. To remove this bias, we reduced the number of images in each folder to approx. 4000.
4) Original source, that was parsed, extracted and modified is CROHME dataset.
5) The CROHME dataset provides more than 10,000 expressions handwritten by hundreds of writers from different countries, merging the data sets from 3 CROHME competitions.
6) Different devices have been used (different digital pen technologies, white-board input device, tablet with sensible screen) so different scales and resolutions are used.

CHAPTER 6

LITERATURE SURVEY

Some challenges connected to the topic of online mathematical expression recognition were examined by Ahmad-Montaser Awal et al (2010). Ha. et al. (1995, August) devised a system that can deduce mathematical expressions from the pictures of printed documents. They used 5 object-oriented methodology to explain the data abstraction for the hierarchical structure of the mathematical expression, which is presented in the form of an expression tree, in the development of this system. Using a feed-forward neural network approach, Pradeep et al. (2010) suggested a diagonal feature extraction technique for the handwritten character. For categorization, this technique employs diagonal, horizontal, and vertical features. A few papers on the recognition of mathematical expressions using convolutional neural networks are available online (CNN).

Azzeddine Lazrek, Widad Jakjoud International Conference on Multimedia Computing and Systems (ICMCS), 2011. "Segmentation approach of offline Mathematical symbols." The goal of this paper is to identify, extract, and segment the various mathematical symbols. Later on, this expression will be recognized.

Zouaoui Abderaouf 2014 global conference on computer applications and research "licence plate character segmentation based on horizontal projection and linked component analysis" (wscar). A license plate segmentation method for Algerian cars is proposed in this paper. The proposed system is separated into two parts: first, the license plate is identified from the input image, and then the characters from the license plate are segmented.

Catherine Lu Karanveer Mohan "Recognition of Online Handwritten Mathematical Expressions Using Convolutional Neural Networks," Catherine Lu Karanveer Mohan cs231n project report Stanford 2015. We delve more into the challenge of identifying handwritten mathematical statements, which we also chose as the subject of our CS221 final project.

Pooja Kamavisdar, Sonam Saluja, Sonu Agrawal “A Survey on Image Classification Approaches and Techniques,” worldwide Journal of Advanced Research in Computer and Communication Engineering, Vol. 2, Issue 1, January 2013 Various categorization strategies are taken into account in this study; Decision Tree (DT), Support Vector Machine (SVM), and Fuzzy Classification are all examples of artificial neural networks.

Nicholas E. Matsakis, The Massachusetts Institute of Technology published "Recognition of Handwritten Mathematical Expressions" in May 1999. In this paper, I will describe an online 6 method for converting a handwritten mathematical expression into an equivalent expression in a typesetting command language such asTEXor MathML, as well as a feedback- oriented user interface that can make errors more tolerable to the end user because they can be corrected quickly.

CHAPTER 7

METHODOLOGIES

4.1 Methodologies Explanation:

First, noise from the original input image is reduced using our proposed method by binarizing it. Then, from the input image, we utilize compact horizontal projection to segment each line of equation. Then, for subsequent processing, we treat each segment of the segmented image as a full image. We then look for certain characteristics in the form of related components for each line of equation image. After that, each segmented character is fed into a convolutional neural network model for character categorization. The resulting character, which is CNN's output, is then utilized to create a character string that looks like the original equation.

4.2 Dataset preparation

The preparation of the dataset is the most important aspect of this project. The borders of characters like the English numeral, alphabet, and mathematical symbol can all be accurately defined. As a result, we begin by preparing the dataset with the highest priority given to its edges, i.e., illumine the edges. We created our own datasets and used a modified version of the NIST dataset for digits, which is similar to the popular MINIST dataset. For the training of the network, we use 2000 data items for each category. And in the majority of cases, our network training was accurate to the tune of 98.5 percent. We used a 32x32 grey level image in our dataset.

4.3 Pre-processing

The procedure of changing and modifying the input image to make it suitable for recognition is known as pre-processing. Image enhancing techniques include the ones listed below.

1) Conversion of RGB to Gray-Scale Because character detection on a coloured image is more difficult than on a grayscale image, this coloured image is first turned into a conventional Gray-scale image and represented through a single matrix. If the grey bitmap is Y and the color bitmaps are R, G, and B, the formula is Y = 0.299R + 0.587G+0.114B.

```
In [3]: def load_images_from_folder(folder):
            train_data=[]
            for filename in os.listdir(folder):
                img = cv2.imread(os.path.join(folder,filename),cv2.IMREAD_GRAYSCALE)
        #         print(img.shape)
                img=~img
        #         print(img.shape)
                if img is not None:
                    ret,thresh=cv2.threshold(img,127,255,cv2.THRESH_BINARY)
                    ctrs,ret=cv2.findContours(thresh,cv2.RETR_EXTERNAL,cv2.CHAIN_APPROX_NONE)
        #             print(ctrs.shape, print(ctrs))
                    cnt=sorted(ctrs, key=lambda ctr: cv2.boundingRect(ctr)[0])
                    w=int(28)
                    h=int(28)
                    maxi=0
                    for c in cnt:
                        x,y,w,h=cv2.boundingRect(c)
                        maxi=max(w*h,maxi)
                        if maxi==w*h:
                            x_max=x
                            y_max=y
                            w_max=w
                            h_max=h
                    im_crop = thresh[y_max:y_max+h_max+10, x_max:x_max+w_max+10]
                    im_resize = cv2.resize(im_crop,(28,28))
                    im_resize = np.reshape(im_resize,(784,1))
                    train_data.append(im_resize)

            return train_data
```

2) Binarization

Binarization is the process of converting pixel data into 0s and 1s by selecting a threshold value. In this study, 1s indicate black pixels and 0s represent white pixels in the horizontal projection computation. Binarization thresholds can be approved in two ways: overall threshold and partial threshold. Otsu's method is based on picture statistical properties and is 11 an overall threshold method. This strategy allows the computer to choose a threshold on its own.

3)Noise Reduction

Noise refers to the presence of too many pixels in an image. Salt and pepper noise and Gaussian noise are two types of noise. Low pass filtering is used to remove Gaussian noise from the image, and Salt and Pepper noise does not need to be filtered because it is relatively low in comparison to Gaussian noise. For the sake of simplicity, we deleted all components that are less than 5 pixels in our proposed solution.

4) Segmentation

Image processing and computer vision applications frequently use segmentation to identify objects or other key information in digital images. Which is the division of one image into several parts. In our proposed method, segmentation is divided into two parts.

5) Loaded Data

Load the dataset

```
In [4]: data=[]
```

```
In [5]: # assign "-" = 10

        data = load_images_from_folder(r'C:/Users/ZEEL/Downloads/data/extracted_images/-/')
        len(data)
        for i in range(0,len(data)):
            data[i] = np.append(data[i],["10"])
        print(len(data))

        33997
```

```
In [6]: #assign + = 11
        data11=load_images_from_folder(r'C:/Users/ZEEL/Downloads/data/extracted_images/+/')

        for i in range(0,len(data11)):
            data11[i]=np.append(data11[i],['11'])
        data = np.concatenate((data,data11))
        print(len(data))

        59109
```

```
In [7]: data0=load_images_from_folder(r'C:/Users/ZEEL/Downloads/data/extracted_images/0/')
        for i in range(0,len(data0)):
            data0[i]=np.append(data0[i],['0'])
        data=np.concatenate((data,data0))
        print(len(data))

        66023
```

```
In [8]: data1=load_images_from_folder(r'C:/Users/ZEEL/Downloads/data/extracted_images/1/')

        for i in range(0,len(data1)):
            data1[i]=np.append(data1[i],['1'])
        data=np.concatenate((data,data1))
        print(len(data))
```

CHAPTER 8

ARCHITECTURE

Convolutional layer, pooling layer, completely connected input layer, fully connected layer, and fully connected output layer are all layers in the CNN design.

1. Convolutional layer: The backbone of any CNN working model is the convolutional layer. This layer is where the images are scanned pixel by pixel and a feature map is created to define future classifications.

2. Layer for pooling: Pooling is also known as data down sampling, in which the total dimensions of the photos are reduced. Each feature's information from each convolutional layer is condensed to only include the most essential data. The creation of convolutional layers and the use of pooling is a continuous process that may require multiple iterations.

3. Fully connected input layer: The flattening of the images is also known as the fully linked input layer. The previous layer's outputs are flattened into a single vector.

4. Fully connected layer: When it's time to compute after the feature analysis, this layer applies random weights to the inputs and predicts a suitable label.

5. Output layer: The CNN model's final layer stores the results of the labels determined for classification and assigns a class to the images.

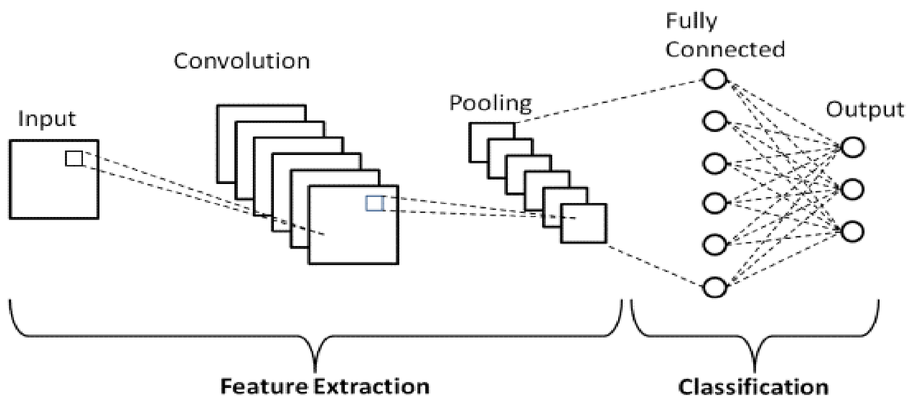


Fig -1: Feature Extraction and Classification of CNN

In [14]:
```
model = Sequential()
model.add(Conv2D(30, (5, 5), input_shape=(28, 28, 1), activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Conv2D(15, (3, 3), activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.2))
model.add(Flatten())
model.add(Dense(128, activation='relu'))
model.add(Dense(50, activation='relu'))
model.add(Dense(13, activation='softmax'))
# Compile model
model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
```

In [15]:
```
from tensorflow.keras.models import model_from_json
```

In [16]:
```
model.fit(x=l, y=cat[:47504], epochs=10, batch_size=32,shuffle=True,verbose=1)
```

```
Epoch 2/10
1485/1485 [==============================] - 8s 6ms/step - loss: 0.0063 - accuracy: 0.9987
Epoch 3/10
1485/1485 [==============================] - 8s 6ms/step - loss: 0.0040 - accuracy: 0.9991
Epoch 4/10
1485/1485 [==============================] - 8s 6ms/step - loss: 0.0057 - accuracy: 0.9988
Epoch 5/10
1485/1485 [==============================] - 9s 6ms/step - loss: 0.0042 - accuracy: 0.9990
Epoch 6/10
1485/1485 [==============================] - 11s 7ms/step - loss: 0.0038 - accuracy: 0.9994
Epoch 7/10
1485/1485 [==============================] - 9s 6ms/step - loss: 0.0035 - accuracy: 0.9994
Epoch 8/10
1485/1485 [==============================] - 9s 6ms/step - loss: 0.0022 - accuracy: 0.9995
Epoch 9/10
1485/1485 [==============================] - 9s 6ms/step - loss: 0.0046 - accuracy: 0.9994
Epoch 10/10
1485/1485 [==============================] - 10s 7ms/step - loss: 0.0024 - accuracy: 0.9995
```

Out[16]:
```
<tensorflow.python.keras.callbacks.History at 0x2aafb0bce80>
```

CHAPTER 9

TECHNOLOGY

Data is collected first in our implemented approach. The data was then standardized. Normalization is divided into two parts: training data and testing data. The training data is then fed into a Convolution Neural Network, which treats each portion of the image as a complete image for processing. We then look for certain characteristics in the form of related components for each line of equation image. Each segmented character is then sent into a Convolutional neural network model, which is used to classify the character. The character obtained as a result of CNN is then utilized to create a character string that is similar to the original equation. The correctness of these characters is then checked. The answer is predicted by the prediction model, which is then delivered as an output.

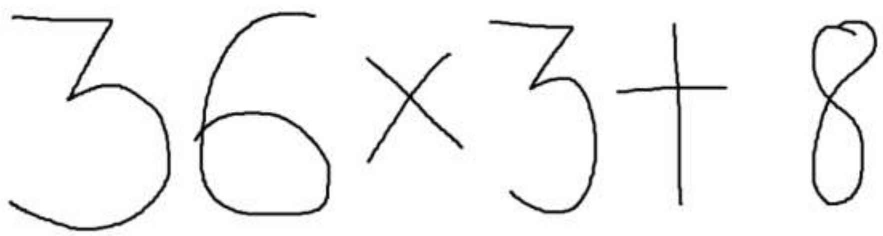
TESTING IMAGE

PREDCITED OUTPUT

```
[10] from IPython.display import Image

     print("\n"*2, "The test image -->")

     Image('image_2.png')
```



CHAPTER 10

RESULT

```
The evaluation of the image gives equation :  36*3+8

/usr/local/lib/python3.7/dist-packages/tensorflow/python/keras/engine/sequential.py:450
  warnings.warn('`model.predict_classes()` is deprecated and '
```

```
print("\n"*2, "The evaluation of the image gives --> ", s, " = ", eval(s), "\n"*2)
```

```
The evaluation of the image gives --> 36*3+8  =  116
```

CHAPTER 11

CONCLUSION

The above presents some ideas on methods of building new handwriting recognition systems and ways of stretching their limits. However, as pointed out at the beginning of this paper, in order to build more sophisticated numeral recognition systems and incorporate more knowledge, all stages of the recognition process—pre-processing, feature extraction and classification--must be considered again. In our view, the most important problem to address is feature selection and extraction. In many publications, there is often little information on how features are extracted and there is generally no specific assessment of how well the selected features are extracted and located by the proposed method. Furthermore, when the system is finally tested, the limitations of its feature selection and extraction are rarely considered explicitly as a cause for the observed rejections and misclassifications. In and, we have begun a more systematic examination of the problem of extracting curvature features from the contours and skeletons of handwritten characters.

This paper has provided some answers to the above questions. It contains a description of our current thinking on ways of stretching the limits and building a new generation of current handwriting recognition systems. We welcome any suggestions you may have to help us reach the ultimate goal of making computers outperform humans in this challenging subject.

CHAPTER 12

FUTURE SCOPE

In future days the main focus will be to try to raise the precision level and build a segmentation system that can successfully segment two connected digits, and also increase the performance level of the dataset.

CHAPTER 13

REFERENCES

1. Chuangxia, H.; Liu, B. New studies on dynamic analysis of inertial neural networks involving nonreduced order method. Neurocomputing 2019, 325, 283– 287.
2. Alvear-Sandoval, R.; Figueiras-Vidal, A. On building ensembles of stacked denoising autoencoding classifiers and their further improvement. Inf. Fusion 2018, 39, 41–52.
3. Zeng, D.; Dai, Y.; Li, F.; Sherratt, R.S.; Wang, J. Adversarial learning for distant supervised relation extraction. Comput. Mater. Contin. 2018, 55, 121– 136.
4. Cai, Z.W.; Li-Hong, H. Finite-time synchronization by switching state-feedback control for discontinuous Cohen– Grossberg neural networks with mixed delays. Int. J. Mach. Learn. Cybern. 2018, 9, 1683– 1695.
5. Xiang, L.; Li, Y.; Hao, W.; Yang, P.; Shen, X. Reversible natural language watermarking using synonym substitution and arithmetic coding Comput. Mater. Contin. 2018, 55, 541– 559.
6. Qu, X.; Wang, W.; Lu, K.; Zhou, J. Data augmentation and directional feature maps extraction for in-air handwritten Chinese character recognition based on convolutional neural network. Pattern Recognit. Lett. 2018, 111, 9– 15.
7. Wang, D.; Lihong, H.; Longkun, T. Dissipative and synchronization of generalized BAM neural networks with multivariate discontinuous activations. IEEE Trans. Neural Netw. Learn. Syst. 2017, 29, 3815– 3827.
8. Kuang, F.; Siyang, Z.; Zhong, J.; Weihong, X. A novel SVM by combining kernel principal component analysis and improved chaotic particle swarm optimization for intrusion detection. Soft Comput. 2015, 19, 1187– 1199. 13. Ciresan, D.C.; Meier, U.; Schmidhuber, J. Multicolumn deep neural networks for image classification. 24 arXiv 2015, arXiv:1202.2745