Comparative Analysis of Convolutional Neural Network and Character Recognition Techniques for Handwritten Mathematical Equation Solver

[1]Navaneetha Krishnan M, [2]Swetha G, [3]Saravanakumar R, [4]Nandhinishri S.N, [5]Arthi R,
[1]Assistant Professor, [2,3,4,5]Scholar, [1,2,3,4,5]Department of MCA,
[1,2,3,4,5]Karpagam College of Engineering, Coimbatore, India.
[1]navaneethanmca@gmail.com, [2]swethagovindharaj10@gmail.com, [3]saravanaanutha06@gmail.com,
[4]nandhinishri01@gmail.com,[5]arthikrish28@gmail.com

***Abstract*—** Solving mathematical equations manually can pose various difficulties, depending on the complexity of the equation and the level of mathematical knowledge and skills of the individual like Lack of understanding, arithmetic errors, incorrect assumptions, and complex equations. This process involves recognizing handwritten characters, which is a difficult task digitizing handwritten text due to a variety of factors. This project aims to develop a system that can automatically identify and solve handwritten mathematical equations using the CROHME dataset by CNN and LeNet character recognition technique. The CROHME dataset contains handwritten characters, including mathematical symbols, which will be used to train a CNN. The LeNet architecture was one of the earliest successful applications of deep learning for image acknowledgement and has since become a classic model in the arena of computer vision. The system will take an image of a handwritten mathematical equation as input and use the trained CNN to recognize the characters and symbols in the equation. Once the equation is recognized, it will be solved using mathematical operations, and the output will be displayed. The results of this project have the potential to be used in educational and professional settings to help individuals solve mathematical problems more efficiently. In experimental analysis, the comparative analysis was conducted by both learning and deep learning algorithms such as CNN, SVM, Decision Tree, and Naive Bayes, and character recognition techniques such as LeNet, OCR, VGG, and ResNet. However, the CNN algorithm gives a better performance rate with an accuracy of 97.56%, specificity of 96%, sensitivity of 96%, and F-score of 97% respectively. The LeNet character recognition technique gives better performance and it produces 98% accuracy compared with OCR, ResNet, and VGG.

***Keywords*—** Character segmentation; Character recognition; Mathematical Equation; Deep Learning; CNN; Image processing; Polynomial.

I INTRODUCTION

In the modern world, we live in now, it is important to work without paper. Writing, both printed and written by hand, has always been an important way for people to communicate and is used in almost every aspect of their lives. It is used to keep and store information. Because of this, people have always tried to find ways to keep it going from one generation to the next. In fact, with the rise of new info technologies like microchip technology and computers and the upsurge in the power of machines, it seems inevitable that processing tasks like reading, researching, and storing will be done by machines [1]. Investigators in the area of form recognition, which includes automatic writing recognition, are worried about it.

This is how research in this arena started in the last few periods.

Machine-readable text from printed or handwritten text is the goal of automatic writing recognition, a complex computer procedure. One way to accomplish this is to transmit the text to the machine for interpretation. Being able to read and write simplifies any and every work involving the whole management of paper documents. As a result, it encompasses a wide range of applications with massive databases, including those that automatically update administrative files, sort mail, read amounts and bank checks, process postal addresses, process forms, provide interfaces devoid of keyboards, analyse handwriting, index libraries' archives, read legacy documents, and search databases for specific information [2].

The rapid advancement of computer and internet technology has resulted in most documents, books, and literature in computer science and other fields becoming digital. [3]. Nearly medicine, economics, and more use mathematics. Analyzing and understanding digital documents is a major research concern today[4],. With LeNet English characters and numbers can be recognized more accurately in electronic books[5]. One of the most inspiring tasks in vision is the credit of handwritten mathematical expressions.[6]. While the nesting assembly is two-dimensional and the sizes are different, the segmentation and recognition rate is insufficient. To recognize mathematical expressions, it is necessary to segment the characters and then classify them. During character segmentation, an image of an arrangement of characters is subdivided into separate symbols [7]. In the segmentation of characters from an image, histogram-grounded projection is commonly used; it is also useful in some detection stages.[8]. As a classification model in image processing, CNN is commonly used [9]. Handwritten mathematical equation solver using CNN is a computer vision application that uses deep learning techniques to recognize and solve mathematical equations written by hand. The system is designed to identify individual symbols and characters from an image of a handwritten equation, and then use that information to solve the equation. Specifically created for image identification tasks, the CNN architecture is a sort of neural network. [10]. It consists of multiple layers of neurons that use convolutional filters to extract features from images[11]. In order to do classification or regression, these features are first sent through pooling layers to lower their dimensionality, and then to fully connected layers.

To create a handwritten mathematical equation solver using CNN, the system must be trained on a large dataset of handwritten equations. During training, the CNN learns to recognize individual symbols and characters, such as digits, operators, and variables. Once the CNN is trained, it can be used to recognize symbols in a new handwritten equation and then use that information to solve the equation [12]. The development of a handwritten mathematical equation solver using CNN has many practical applications, including the ability to digitize handwritten equations for use in digital systems, assist in the grading of handwritten mathematical exams, and provide assistance for individuals who have difficulty with handwriting due to disabilities or injuries [13]. The first signs of writing date back to about 3300 BC, which is a few years. This is a short amount of time compared to how long it took for people to appear [14].

Writing's appearance is a symbol of the change from prehistory to history. It also shows that human societies are changing quickly and going through big changes. As civilizations have grown, it has become important to find a technique to keep the laws and to share and pass on the results of the work of scientists at the time. Archeologists have found the first signs of writing in Mesopotamia, the Levant, Egypt, Persia, and ancient Greece. Before the Chinese conceived paper in the second century, this information was first passed along on tablets made of clay and stone. There is a lot of numerical information that could be useful in a variety of computing tasks, but it is not available in an electronic format. The goal of the mathematical equations identification framework is to turn any systematic paper into a digital file. This framework has been a research topic for many years, but there are still a lot of problems with it.

A. Organisation of this paper

The rest of this paper is satiated as shadows: Part II covers a presentation of the literature review. In Part III, the projected system is described. Section IV examines the approach. Results are obtainable in Part IV, and the conclusion is obtainable in Section V.

II LITERATURE REVIEW

Known by his pen name, Rashad Al-Jawfi. One of the most prominent test beds for neural network methods, character credit is now among the most fruitful uses of this technology. In this study, we provide a novel network for identifying a specific collection of handwritten Arabic letters. There are two phases to this brand-new network. The first step is to identify the character's overall form, and the second is to identify its individual dots. There is also a presentation of the network's structure, properties, and training algorithm[15].

It was Khanh Minh and company. In this paper, we present an incremental documentation strategy for online handwritten mathematical expressions, which allows for an occupied identification interface with little waiting time (MEs). They use ongoing strokes and focus on deft skills at the neighbourhood level. To take into consideration the most recent stroke, one must undertake segmentation, tagging, and a recalculation of the most up-to-date Cocke-Younger-Kasami (CYK) table. With a multi-threaded strategy, we were able to cut down on waiting times. Our data set demonstrates that the incremental method is effective, with almost the same recognition rate as the group recognition method and significantly less waiting time. A middle ground between the two is also proposed. By refreshing the geometric features, identifying new competitors, and refreshing the CYK table after every approved stroke, the suggested strategy decreases waiting time while keeping the same pace as the batch method. Our experiments on a dataset consisting of 2,958 handwritten mathematical equations proved this to be true. There is no difference in success rate between the two strategies when combined, and the waiting time is cut by a factor of 39 compared to the incremental approach, according to the research[16].

According to the work of Cuong Tuan Nguyen et al., This article demonstrates how to segment manually written online information in English using Bidirectional Long-Short Term Memory repeating neural networks. Using the systems, you can mix and match long-term forward and backward settings to make sure that segmentation is more robust than vulnerability. Using this method to partially accept online English material written by hand reduces waiting time by up to 62% and processing time by 50%, as shown. The recognition rate of the framework also improves by 3%, going from 71.7% to 72[17].

For the purpose of distinguishing between handwritten and machine-printed text, Mehryar Emambakhsh et al. developed a template-matching approach. As a first step, you must pre-process the scanned papers' images by eliminating distracting background elements, obscuring unnecessary shapes like circles and lines, and grouping similar words into larger blocks. Then, mix and match the characters in an adaptable gallery with the various components utilising parallelized standardized cross-correlation. The system excels at classifying samples that are muddled, blocked, noisy, or lacking in essential information, as seen by its performance on the Pattern Recognition and Visual Analysis Research Lab-Natural History Museum dataset. For the entire data set, the calculation yields an order rate of 84.0% and a false positive rate of 0.16. It requires no prior test preparation and outperforms similarly prepared approaches on the same benchmark with significantly more compelling results[18].

This research proposes a method for distinguishing between handwritten and machine-printed text. In order to tackle the HMC problem, a presentation of machine-printed samples is supplemented by a method of identifying machine-printed samples, as the handwritten text has many more variations. Word-block level segmentation is performed after the images of scanned documents have been pre-processed in the background, including denoising and the removal of circles and lines. It turns out that the proposed strategy works quite well at de-noising and re-ordering samples that are otherwise disorganized, obstructed, loud, and/or lacking in crucial details. Unlike traditional approaches, the proposed method does not require training, which eliminates the need for data samples on which a classifier can be trained. Despite the fact that the algorithm projected is generic. The paper's suggested algorithm has room for improvement, especially with the addition of a phase to identify text style and the implementation of a more robust alignment algorithm for overlapping text. False positives for the machine-printed class can be reduced and the time it takes to identify the proper template in the gallery can be shortened if the detection of 40 text styles is successful. The algorithm's capacity to disentangle the overlapping samples can be put to the test in further detail using even more challenging datasets.

According to Hubert Cecotti,it is still difficult to recognize handwritten numbers, especially in texts with a variety of writers with notably diverse writing styles. This dynamic range is an intriguing challenge for pattern recognition and image processing algorithms. The availability of powerful personal computers, lightning-fast networks, and sophisticated software for distributed computing has made it possible to apply computationally intensive techniques to massive data sets. The goal of this work is to investigate the effect of various parameters and pre-processing techniques on the precision of an image restoration-based separation. One major issue with nearest-neighbor categorization is the length of time it takes to process. Dismissal guidelines and adjustable distances are a good place to start. There are three databases of handwritten numbers from India where they hope to gauge the performance of single-character recognition. Every one of these archives is dedicated to one of three widely spoken Indian languages: Bangla, Devnagari, and Oriya. Eliminating features associated with the cardinal directions significantly enhanced accuracy, they demonstrated[19].

Chen Liu et al.,It is more difficult to extract equations from photographs of documents saved as images due to the widespread availability of PDF documents. In this research, we present a method for automatically detecting and extracting mathematical formulas from PDF images of English-language documents, which is a necessary step towards a solution to the issue. Segment selection, character block separation, and character block assembly are the three main components of the design. This study presents a related parameter-adjustment approach for shielding the implementation of mathematical expression identification from the negative impacts of resolution changes. To do this, we dissect PDF document images, detailing their characteristics and the impact they have on the recognition of mathematical expressions. The exploratory research reveals that some use expands the adaptability of the algorithm for identifying mathematical formulae[20].

Neural networks, as discovered by Soni Chaturvedi et al. are an excellent resource for neuron to demonstrate the recognition of digits and special characters. Digits and punctuation are used to create the input signal. As soon as that happens, the Izhikevich neuron model and the feedforward neural system kick in to determine the firing rates. If the neural model's synaptic weights and threshold point digits are modified, the input pattern can be observed while maintaining a nearly constant firing rate. Finally, a comparison is done in MATLAB between the two models of Spiking Neural Networks, the Izhikevich neural system, and the feed-forward neural system, for manually written Pattern detection[21],.

The recognition of hand-written digits is a significant topic of study in computer vision and pattern recognition, according to Caiyun et al. In this research, we propose a method for recognising handwritten digits that is based on the careful extraction of many aspects. Images with varying aspect ratios and stroke thicknesses required initial normalization in pre-processing to eliminate unwanted data while preserving relevant elements. You need also to define features like structure feature, distribution feature, and projection feature, as manual digit recognition differs from conventional digit semantics recognition. The deep neural systems used for semantics identification should also incorporate a wide variety of factors. Tests demonstrate that the proposed algorithm performs admirably and is more widespread than a select group of competing methods. [22].

According to Rana S. et al. (OCR) is simply the text. This method of data entry sees the widespread application. Classifiers employed in this approach were

ANNs. This study proposes a system for offline OCR that can read handwritten Arabic letters. The ANN was built using the Hopfield Calculation, which was designed in MATLAB. There is an initial step of "pre-processing," then "feature extraction," and ultimately "identification" of the image. Several characteristics of each letter, often known as "features," are calculated and then removed from the picture to ensure precise recognition. The most crucial is the selection of characteristics from the image. The number of these features (called vectors) can tell an ANN a lot about a person's personality. The experimental findings demonstrated that the framework correctly identified eight of 42 handwritten Arabic letters (77.25%). Changes to the framework's structure can add a segmentation phase and allow it to comprehend the remaining letters of the Arabic script [23].

As the summary of the literature review, Concerned the work is making and approving an automatic system that can read handwritten math equations made up of letters and symbols. The system's performance is measured by how often it works, how well it works, etc. Calculating the number of correctly recognised equations from a set of given equations gives the percentage of accuracy. The main things that are looked at are segmentation, classification, and the ability to recognise an equation. Segmentation is the process of breaking up an image into separate characters or objects. This is done by grouping pixels that belong to the same symbol, finding the center of each symbol, and using a labeling procedure to give each segmented symbol a name. Different types of equations, such as simple equations, complex equations, etc., are put into different groups. The best classifier is the Convolutional neural network. To recognise a math equation, you have to put the symbols in the order they appear in the equation. This lets you put the equation back together and recognise it. Even though a lot of people have worked on math equations and symbol recognition, there are still a lot of problems that need to be solved. These problems include separating the symbols from the equation, improving the recognition rate, recognising symbols or characters that are touched or written over, recognising complicated equations, making the system more efficient overall, etc.

III PROPOSED SYSTEM

The proposed deep learning-based handwritten equation solution Users might enter handwritten equations into a system like Python, which would then produce the solved solution. Convolutional neural networks (CNNs) would be used by the system to recognize and understand the inputs, and deep learning algorithms would be used to solve the problem. The deep learning method would be created to solve equations based on the input photos, and the CNN would be trained on a collection of images, each of which represented an equation. On the user's screen, the solution to the problem would be displayed. Additionally, the system would make use of an input pre-processing module to change the input into a format that the CNN could understand. This module would be used to normalize the size of the equation and remove any noise from the input. The input would be given to CNN for recognition and interpretation following pre-processing. The deep learning algorithm, which is intended to solve the equation based on the input, would then receive the output of the CNN. The user would then be shown the solved equation, which is the algorithm's output. The suggested technique would provide a quick and accurate method for resolving handwritten equations. It would make a fantastic teaching tool for both teachers and students.

The proposed method aims to solve Handwritten Mathematical Equations using CNN and LeNet techniques and compare its performance with the most popular algorithm such as SVM, Decision Tree, Random Forest, and Naive Byes and Compare LeNet character recognition technique with other similar techniques such as OCR, VGG, and ResNet.

A. Database

People's writing varies in terms of size, font, stroke width, and style. Data collected from members of society across the lifespan must be taken into account. A variety of data types, including the integers 0 through 9, letters, and symbols used in the alphabet and other writing systems, and even more obscure data types, are taken into account. All the data in this database—numbers, symbols, and characters—follow a strict sequential format, with each element in the series given its own row and column. The CROHME dataset is a benchmark dataset for online handwritten mathematical expression recognition. The dataset was created to support research on the development of accurate and efficient systems for recognizing handwritten mathematical expressions in real time.

The CROHME dataset includes over 3,000 handwritten mathematical expressions, consisting of symbols, digits, and operators. The expressions were collected from a variety of sources, including contributions from volunteers, and covered a range of mathematical notations, including integral signs, fractions, and matrices. The dataset is fragmented into a training set of 1,000 expressions, a validation set of 500 expressions, and a test set of 1,500 expressions, with a balanced distribution of expressions in each set. The expressions are in vector format and have timing information, representing the order and timing of the strokes used to write the expression. Several benchmark results have been published on the CROHME dataset, with state-of-the-art accuracy achieved using deep learning models.

The CROHME dataset has been widely used in research on online handwritten some expression recognition, including studies on sequence-to-sequence models, attention mechanisms, and multi-modal approaches that combine handwriting and speech signals. The database is shown in figure 1 as follows.

Figure 1: Database samples [24]

B. Character Recognition

In the last 50 years, work on character recognition has grown steadily and reached a point where it can be used in post offices, banks, and other government and non-government offices to identify characters and make it much harder to make mistakes when reading. LeNet is the process of making systems that can turn typed or handwritten text into a format that a computer can read. At first, OCR was only used to recognize characters, but as technology has improved, a lot of progress has been made in this area. Computer-based pattern recognition and vision communities study a lot of things like word recognition systems, even if it's just text, a diagram, or a math equation. This is a misnomer, but it makes things easier.

Character recognition is a broad term for a number of different ways that machines can recognise characters in different fields of use. It has growing needs in areas that have been growing recently, such as the development of electronic libraries, interactive media information bases, and systems that need the information to be written by hand. It can be ordered based on real-time processes like data collection (online or offline) and text writing (machine-printed or manually written).

People have used a special 2-D notation to talk about math with each other for more than a hundred to two hundred years. The notation is made to show how it helps with both thinking about math and seeing it. Using this same two-dimensional notation to talk to PCs is natural and the best way to do it. This includes communication between the two-dimensional notation used in math and the internal representations used by the PC. With the technology we have now, PCs can create two-dimensional mathematical notations like symbols, digits, etc., but there aren't many ways to read them. Most of the time, it's a human user's job to turn an interpretation of math into a form that a computer can understand. With the number of Internet users growing so quickly these days, there is a growing trend of spreading and exchanging information through this well-known channel. Because of how much people use the Internet, advanced libraries and distance learning are becoming popular places to do research. This is because the Internet is used so much that location problems have arisen. One of the most important ways to understand these ideas is to come up with simple and effective ways to translate information from paper records into an electronic format that can be read by computerized PCs and sent over the Internet.

Mathematical identification is a crucial first step in many branches of science and engineering. In order to use all the keys on a computer, it is common to practice altering the format of mathematical equations. Alternatively, you can utilize a series of alternative key combinations to communicate with other special symbols by pressing a combination of regular keys and a few extra keys (such as functional keys) in the console. Given the complexity of the underlying structure, identifying mathematical equations has been and continues to be a hot area of study. These technologies can be used to digitize scientific reports, recover lost data, or help the visually impaired interpret mathematical and other numeric expressions, all of which have a significant impact on the field of scientific archive image analysis. Humans have contributed mathematical notation, handwritten equations, and other forms of a mathematical expression to this framework. Scientific writing, including mathematical equations, is far more difficult to enter into a computer than regular languages, such as letters and words. This is because numbers, symbols, touching characters, distinct symbol fonts, overwriting, complex symbols, digits, and letters all have their own administrators and particular meanings in scientific writing.

For an online approach, the framework uses data about strokes that are entered in real-time and change quickly. While the offline method handles math equations that are printed or written by hand, the online method scans the documents and identifies the math equations that were scanned. This framework has to deal with a lot of problems. For example, when it comes to manually written scientific appearance all change from one person to the next.

C . Classification of Symbols

The symbol classifier is the second problem that a 2D language recognition system has to deal with. The most important step is to put symbols into groups, but it doesn't have to be the hardest step. This is because there is a lot of information about pattern recognition.

But the way these 2D languages are made means that there are some things that must be taken into account when making the best recognition system. Handwriting isn't always the same, so the classifier has to take that into account. Different people can write the same symbol in different ways. Symbols should also be able to be written in any order on more than one line. Also, some 2D languages have a very large number of symbols. For instance, mathematical notation has more than 220 symbols. Some of these symbols may have different meanings based on where they can show up in a graph in different sizes and directions. Most of the time, a step of pre-treatment and standardization is needed to make the classification more reliable.

D .Detailed overview of handwritten mathematical equations classification and recognition

Figure 2 shows the whole system architecture for recognizing mathematical equations. It is mostly made up of the recognition stages. Step-by-step explanations of each stage are given below, along with the goals of each stage.

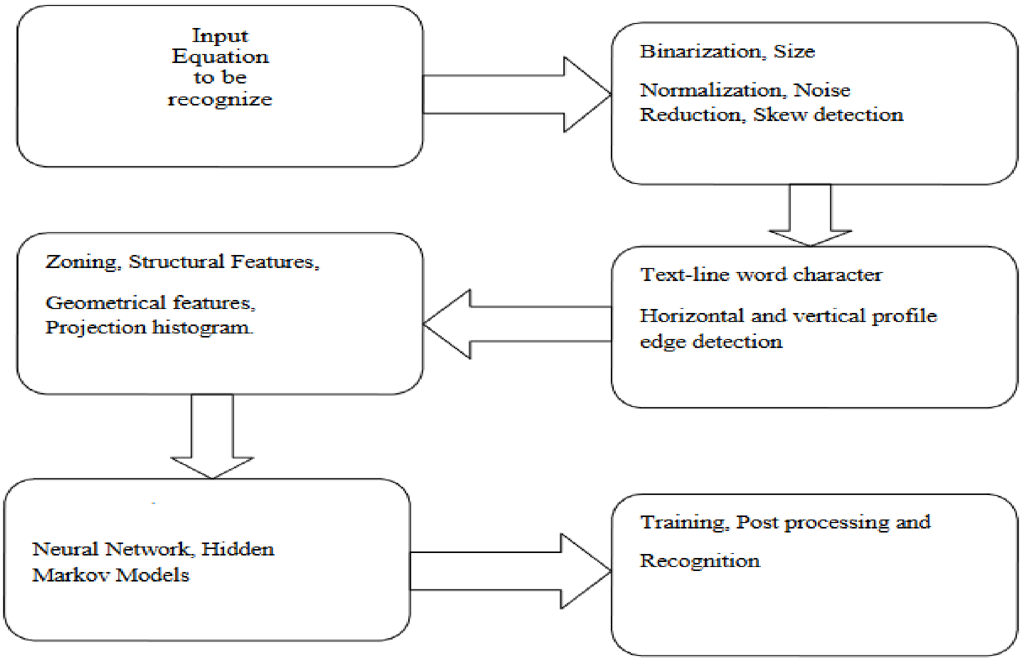

Figure 2: Architecture of mathematical equation recognition system

Steps involved in the project:

1. **Input:** The system receives handwritten images in the mathematical equation as input. The images are captured using a mobile device.
2. **Pre-processing:** The input image undergoes pre-processing, which includes converting the image to grayscale, resizing, and normalizing the image.

2.1 Conversion of RGB to Gray-Scale

Converting the image from RGB (Red, Green, Blue to grayscale involves transforming each pixel's RGB values into a single gray value. This can be done using a formula, which calculates the gray value as the average of the R, G, and B values.

$$Gray = 0.299R + 0.587G + 0.114B \quad (1)$$

Where R, G, and B are the red, green, and blue color channels of the RGB image, and the coefficients 0.299, 0.587, and 0.114 represent the relative luminance of each color channel.

This formula assumes that the human eye is most sensitive blue light. Therefore, the green channel is given the highest weight, followed by the red channel, and then the blue channel. The result of the formula is a single grayscale value that represents the brightness of each pixel in the image.

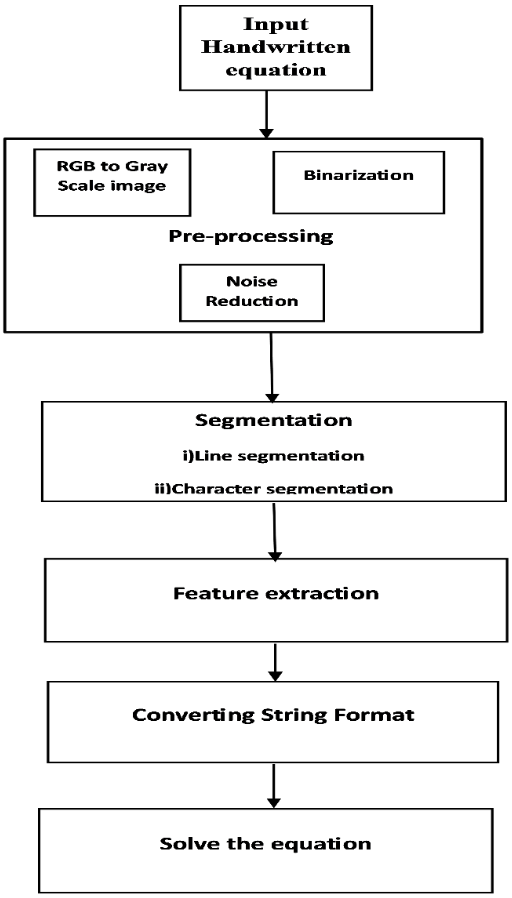

Figure 3. Flow diagram of Mathematical equation solver system

2.2 Binarization

During the process of binarization, a grayscale or color image is transformed into a binary image, where each pixel is either black or white. Binarization aims to simplify an image into its most essential features, making it easier to process and analyze. Several methods for performing binarization include global thresholding, local thresholding, and adaptive thresholding. Global thresholding involves converting an image into binary format by comparing each pixel value to a fixed threshold value. The pixel is set to white if its value is above the threshold; else, it is set to black. Local thresholding involves dividing the image into smaller regions, known as windows, and computing a threshold value for each region. This allows the threshold to adapt to local changes in the image intensity. Adaptive thresholding is a variation of local thresholding, where the threshold is computed based on the mean or median intensity of the surrounding pixels. This method is useful for images with varying illumination or for removing noise from an image.

Binarization is widely used in image processing, particularly in document analysis and image segmentation. It is also used for image recognition tasks, such as LeNet, and for enhancing images for further processing [13]. The binarization formula used in image processing involves comparing each pixel in the input image to a threshold value and assigning a value of 0 or 1 to the pixel based on whether it is below or above the threshold value, respectively. The formula can be expressed as follows:

If $I(a, b) > T$, then $B(a, b) = 1$ (white) (2)

If $I(a, b) <= T$, then $B(a, b) = 0$ (black)

Where $I(a, b)$ is the brightness value of the pixel at location (a, b) in the input image, T is the threshold value, and $B(a, b)$ is the corresponding binary value in the output image. Adaptive thresholding is another binarization method that automatically adjusts the threshold value based on the local properties of the image. This method is useful when there is a significant variation in lighting or contrasts across the image. The formula for adaptive thresholding is:

$T(a, b) = k * mean(I(a, b)) – C$ (3)

If $I(a, b) > T(a, b)$, then $B(a, b) = 1$ (white)

If $I(a, b) <= T(a, b)$, then $B(a, b) = 0$ (black)

Where $T(a, b)$ is the threshold value at location (a, b), k is a constant factor that controls the sensitivity of the threshold, $mean(I(a, b))$ is the local mean brightness of the image around the pixel (a, b), and C is a constant offset.

2.3 Noise Reduction

Noise is the term used to describe too many pixels in an image. Gaussian or Salt and Pepper noise are both examples of noise. Gaussian noise is removed from the image using low pass filtering, while salt and pepper noise is not removed because it is much less than the Gaussian noise. In order to simplify the removal of minor undesired pixel noise, we eliminated all components in our proposed technique that are less than 5 pixels.

3. Segmentation

Segmentation is a common procedure used in image processing and computer vision applications to recognize objects or other significant information in digital images. Segmenting an image into various pieces is necessary.

In suggested strategy, there are two basic steps to segmentation.

3.1 Line segmentation

3.2 Character segmentation

3.1 Line Segmentation

In Convolutional Neural Networks (CNNs), line segmentation divides a document image into individual lines or text segments. This is a crucial pre-processing step for applications like LeNet and document analysis.

Line segmentation's major objective is to precisely identify the text lines in a picture and distinguish them from the background and other objects [25]. This is usually done by applying image processing techniques such as thresholding, morphological operations, and contour detection to the input image.

By directly learning features from the input image, CNNs have recently been utilized to increase line segmentation accuracy. This is done by training a CNN on a dataset of document images, where the desired output is a binary mask indicating the location of the text lines in the image.

A segmentation of equations refers to the separation of different lines in an equation character existing in the image. The vertical gap between characters on one line and those above and below defines each line well. Using this gap, characters not on the same line can be detected and separated. As mentioned above, we use compact horizontal projection for the solution of the square term in quadratics. The threshold value obtained by averaging the maximum value of each horizontal projection vector is used to combine two curves into one representing one equation line with two curves.

Algorithm 1: Dense Horizontal projection

Step 1: Compute M horizontal projection vectors, Y.

Step 2: Determine each vector's extreme.

Step 3: Calculate the Threshold,

$$T_h = \frac{\sum_{i=0}^{M} Max(Yi)}{M} \quad (4)$$

Step 4: Syndicate the curve with the following curve for segmentation for each curve if Max(Y) <T_h.

Once the CNN is trained, it can be used to perform line segmentation on new images by passing them through the network and generating a binary mask as output. The CNN can also be fine-tuned on a specific dataset to improve its performance for a particular application.

3.2 Character segmentation

Character segmentation in Convolutional Neural Networks (CNNs) is the task of breaking down an image of text into individual characters. This is typically a pre-processing step in LeNet systems, which use machine learning algorithms to recognize the characters and convert them into machine-readable text. In a typical character segmentation pipeline, an input image is first passed through a CNN to extract features from the image. These features are then used to identify individual characters and segment them from the rest of the image. This can be done using a variety of techniques, such as region proposals, thresholding, or connected component analysis.

The choice of CNN architecture for character segmentation depends on the type of input images and the requirements of the LeNet system. For example, for handwritten text recognition, a CNN with a large receptive field and a high level of abstraction may be preferred, as it can capture the variability in the shape and size of characters. On the other hand, for machine-printed text recognition, a shallower and more specialized CNN may be more appropriate, as it can take advantage of the regularity and structure of the text.

Once the characters have been segmented, they can be fed into a LeNet system for recognition. In some cases, the character segmentation and recognition can be combined into a single end-to-end system, where the CNN outputs the recognized characters directly.

3. Convolutional Neural Network: Both the training and testing phases of this study employ the same 32x32 grayscale image as the input to the CNN's input layer. The categorization model is a network of convolutional neural cells. A 28x28 feature vector is generated for each input image following the convolution of the input image with the 7x7 filters at the convolutional layer. Adding a nonlinear layer (also known as an activation layer) is acceptable is acceptable to add a nonlinear layer (also known as an activation layer) right after a convolutional layer. The persistence of this layer is to introduce nonlinearity to a system[26] The input is divided into rectangular pooling regions, and down sampling is carried out by calculating the maximum of each zone using a max pooling layer[27]. Downsampling processes are carried out using pooling layers. Calculation of the output size, U of a pooling layer with input size, X Pooling filter size, S Padding, P size Z has been done by

$$U = (X - S + 2\times P) / Z + 1 \qquad (5)$$

The pooling layer output for our suggested strategy is 14x14 and we utilise a 2x2 pool size. In a dropout layer, input elements are arbitrarily chosen to zero with a specified probability. It is an easy method to stop the neural network from overfitting. Overfitting poses a serious challenge in these networks. Huge networks are also slow to operate, making it more difficult to avoid overfitting by integrating several separate large neural nets' predictions at the test time. Dropout is the method used to overcome this issue. The fundamental concept is to randomly remove units from the neural network during training (along with any connections they may have) [28]. The dropout layer's probability is set to 0.3 in the suggested technique during training. The completely linked layer is the one to which every neuron from the layer before associates. A fully linked layer incorporates all the knowledge that the prior layers acquired while processing the image in order to recognise the larger patterns. A mixture of all the attributes is used to group the images in the final, fully connected layer. The number of classes in the target data is therefore reflected in the Output parameter of the final fully linked layer. Our work has a 14-class output scale to match the 14-class input scale.

The fully connected layer's output is normalised by the convolutional neural network model using an activation function called SoftMax[29]. The SoftMax layer produces positive numbers that add up to one, which can be used as the classification probabilities at the classification layer. The classification layer, which is the top layer, finds the input picture classes using the probabilities provided by the SoftMax activation function and determines the loss by contrasting them with the predetermined ground truth classes.

ALGORITHM

Step 1: Start

Step 2: Load the input images of mathematical equations and their corresponding solutions

equation_images, equation_solutions = load_data()

Step 3: Preprocess the images

#equation_images = threshold(equation_images),

#equation_images = edge_detect(equation_images),

#equation_images = resize(equation_images)

Step 4: Use an LeNet tool to extract the text from the equations

#equation_texts= LeNet_tool.extract_text(equation_images)

Step 5: Convert the extracted text into a suitable format for mathematical processing

#equation_texts = convert_to_math_format(equation_texts)

Step 6: Split the data into training, validation, and testing sets

Step 7: Define the architecture of the CNN

Step 8: Initialize the parameters of the CNN

Step 9: Train the CNN using the training set

Step 10: Evaluate the performance of the CNN using the validation set

val_loss, val_accuracy = model.evaluate(val_texts, val_solutions)

Step 11: End

The algorithm begins by loading input images of mathematical equations and their corresponding solutions. Then, the images are pre-processed by applying various techniques such as thresholding, edge detection, and resizing. Next, a LeNet tool is used to extract the text from the equations, which is then converted into a suitable format for mathematical processing. After that, the data is split into training, validation, and testing sets, and the architecture of the convolutional neural network (CNN) is defined. The parameters of the CNN are initialized, and the model is trained using the training set. The training is carried out for a specific number of epochs, with a particular batch size, and the performance of the model is evaluated using the validation set. Finally, the algorithm ends after evaluating the CNN's performance.

4. **Mathematical Solver:** The classified equation is passed to a mathematical solver to find the solution. SymPy library was used to solve the mathematical equation [30]. SymPy is a Python library that provides symbolic mathematical computation. It can solve algebraic equations, perform calculus, and generate symbolic expressions. SymPy can parse a mathematical expression in string format and convert it into an internal symbolic representation.

It can convert the string "$\int x * 2\, dx$ " into the symbolic expression $\int x^2\, dx$

Step 1: Define the string equation

Step 2: Parse the string equation into a SymPy expression.

expr = sp.parse_expr(string_eq)

Step 3: Integrate the expression with respect to x.

integrated_expr = sp.integrate(expr, 'x')

Step 4: Substitute the value of x into the integrated expression and evaluate it.

Step 5: Print the numeric equation.

SymPy can simplify a given expression by applying algebraic rules, trigonometric identities, and other mathematical rules using trigsimp() function. It can simplify the expression $\sin^2+\cos^2$ into the number 1.

Step 1: Define the trigonometric equation

Step 2: Simplify the equation

simplified_eq = sp.trigsimp(eq)

Step 3: Print the simplified equation

SymPy can solve algebraic equations symbolically using various methods like Gaussian elimination, factoring, and substitution [31]. It can solve the equation "$x^2 + 2x + 1 = 0$" and return the solution-1. SymPy can perform calculus operations on a symbolic expression like differentiation and integration. For example, it can differentiate the expression "x*2 + 2x + 1" and return the expression 2x + 2.

5. Output: The solution to the equation is returned as the final output.

IV RESULTS AND DISCUSSIONS

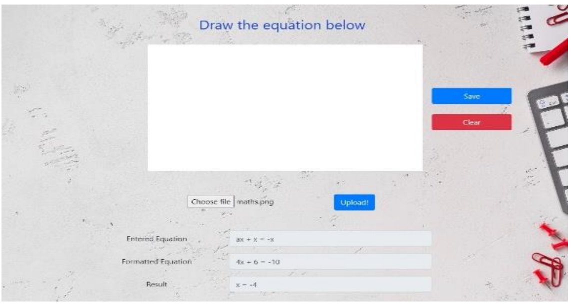

Figure 4. GUI of the handwritten mathematical equation solver by image

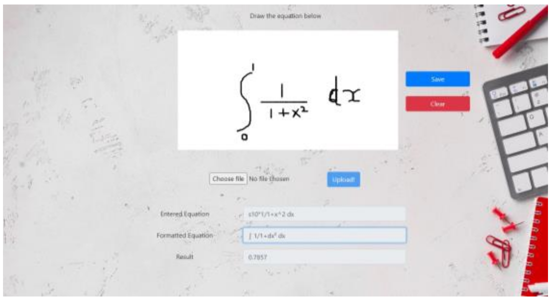

Figure 5. GUI of Handwritten Mathematical equation solver by drawing the equation

Operating System	Windows 7 or Higher
Language	Python 3.7
IDE	PyCharm
Processor	Intel core i3
Hard Disk	200GB
RAM	4 GB or higher

Table 1: Minimum software and hardware requirement

The first stage in image pre-processing is the conversion of RGB to Gray-Scale**.** Each pixel in a grayscale image is represented by a single value, which indicates the brightness of that particular pixel. In contrast, RGB images have three color channels, each representing the intensity of red, green, and blue light at that pixel. By converting an RGB image to grayscale, we essentially collapse these three color channels into a single channel, which makes the image easier and faster to process

TP: True positives (TP) are the sum of correctly classified equations

FP: False positives (FP) are the sum of incorrectly confidential equations

TN: True Negatives (TN) are negative classes that are correctly foretold as negative

FN: False negatives (FN) are the sum of equations that are erroneously classified as negative instances

Predicted labels)	**Algorithm outputs/labels**	
	(Negative/0)	**(Positive/1)**
Mathematical Equation (False/0)	TN	FP
Mathematical Equation (True/1)	FN	TP

Table 2: Confusion matrix for equation forecast

A. Performance Metrics

The proposed model classification performance by using some metrics is discussed in this section.

1. Sensitivity (SE): The number of true positives refers to the symbols that were correctly identified by the model as part of the equation, and the number of false negatives refers to the symbols that were actually part of the equation but were not identified by the model.

$$Sensitivity = \frac{TP}{TP+FN} \quad (5)$$

2. Specificity (SP): In order to avoid developing a Mathematical Equation, the chance of certain outcomes is specified as TN.

$$Specificity = \frac{TN}{TN+FP} \quad (6)$$

3. Accuracy (AC): It's based on the likelihood of getting correctly categorized results as a Mathematical Equation.

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN} \quad (7)$$

4. F-score: The F-score is the mean of the replica's sensitivity and specificity, which is used to assess its performance.

$$F\ score = 2 * \frac{Sensitivity \cdot Specificity}{Sensitivity+Specificity} \quad (8)$$

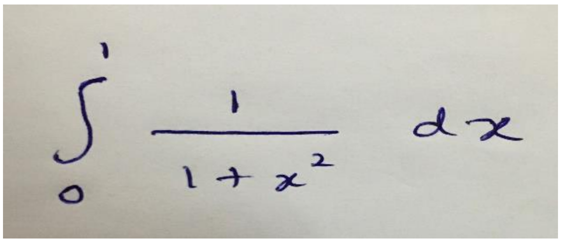

Figure 6. Original equation Image

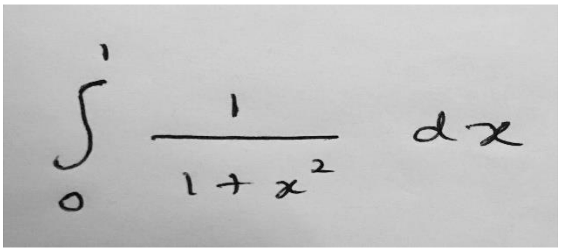

Figure 7. Converted Gray-scale Image

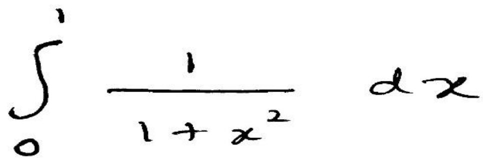

Figure 8. Converted binary Image

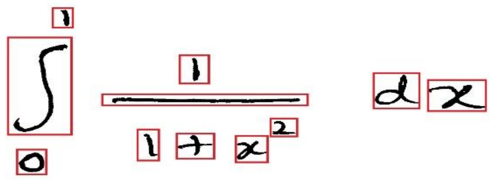

Figure 9. Character Segmentation

B. Performance Analysis of Proposed CNN for different validation

In this section, the validation of the projected method is carried out by changing the percentage of the training dataset and testing dataset. Initially, In the experimental analysis of the proposed method for 70% of training data and 30% of testing data in terms of numerous metrics. CNNs (Convolutional Neural Networks) are particularly well-suited for Handwritten mathematical equation solver tasks due to their ability to automatically learn and extract hierarchical features from images. This is achieved through the use of convolutional layers, which scan the input image with a set of learned filters to identify important features.

Compared to other machine learning algorithms like algorithms, CNNs have several advantages:

CNN is capable of automatically learning features from raw image data, without the need for manual feature engineering. This means that they are able to adapt to a wide range of input variations, including different writing styles and variations in handwriting quality. CNN can handle large amounts of data, making them ideal for complex image recognition tasks like mathematical equation solving. CNN is able to capture spatial dependencies between different parts of an image, which is particularly important for handwriting recognition, where the order and orientation of strokes can vary greatly. CNN is able to handle non-linear relationships between input features, which is important for recognizing complex patterns in handwritten mathematical equations. CNN is particularly suited for image recognition tasks due to its ability to extract local features from images using convolutional layers. SVMs can also be used for image recognition, but they may require hand-engineered feature extraction methods [32].

CNN is well-suited for tasks such as image and audio classification, while Random Forest can work well for a wide range of classification and regression tasks [33]. Handwritten mathematical equations are often represented as 2D images with multiple channels (e.g., grayscale or RGB). CNNs are specifically designed to handle this type of high-dimensional input data, while Random Forest is better suited for tabular data with a limited number of features.

CNNs are particularly suited for extracting features from images, audio, and video data using convolutional layers. Naive Bayes assumes that each feature is independent of all other features, which can lead to poor performance if there are dependencies between features [34]. CNN is a type of neural network that uses convolutional layers to extract features from the input data. Decision Tree is a tree-based algorithm that splits the data into smaller subsets based on the values of the input features [35][36].

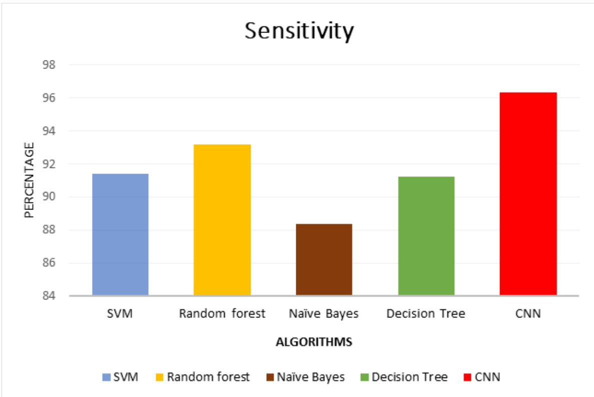

Figure 10: Sensitivity Analysis for SVM, RF, NV, DT, and CNN

The sensitivity of a classifier is defined as the percentage of true positive cases that it properly identifies, and is equivalent to the recall or true positive rate [37]. CNNs can achieve high-sensitivity performance by leveraging their ability to learn complex and hierarchical features from images.

Figure 10 shows that CNN has the highest sensitivity of 96.33%, followed by an SVM model at 91.42%, a Random Forest model at 93.15%, a Naive Bayes model at 88.36%, and a Decision Tree model at 91.25%.

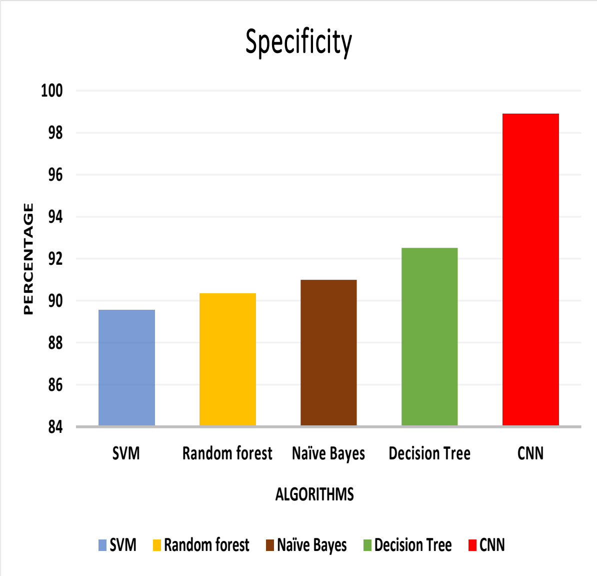

Figure 11: Specificity Analysis for SVM, RF, NV, DT, and CNN

Figure 11 demonstrates that CNN has a specificity of 98.91%, which is higher than an SVM model's specificity of 89.56%, a Random Forest model's specificity of 90.36%, a Naive Bayes model's specificity of 91%, and a Decision Tree model's specificity of 92.52%.

In classification tasks, CNNs can learn to recognize complex patterns and variations in the input data, making them highly effective in detecting subtle differences between classes. This makes them particularly well-suited for image classification tasks, where subtle variations in the shape, texture, and color of an object can be used to distinguish between different classes. In medical diagnosis tasks, for example, CNNs have shown high sensitivity in detecting subtle patterns in medical images that are indicative of certain diseases.

On the other hand, Random Forest, SVM, Naive Bayes, and Decision Tree are machine learning models that are designed to work with tabular data. They are based on statistical and probabilistic methods that are less effective in dealing with high-dimensional data with spatial structure. While they can handle a wide range of feature types and are generally robust to noise and outliers, they may struggle to extract meaningful features from images and other high-dimensional data. CNNs are highly effective in detecting subtle patterns and variations in spatial data, which makes them well-suited for image classification tasks and other high-dimensional data with spatial structure.

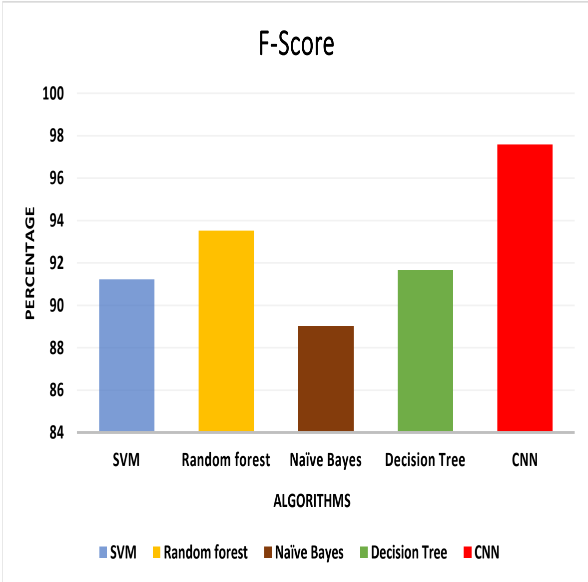

Figure 12: F-score Analysis for SVM, RF, NV, DT, and CNN

In Figure 12, CNN is shown to have the greatest F-Score performance (97.59%), followed by SVM (91.23%), Random Forest (93.52%), Naive Bayes (89.02%, and 91.66%), and Decision Tree (91.23%, and 93.52%, respectively).

CNNs are highly effective in image classification tasks because they can automatically learn spatial hierarchies of features through multiple layers of convolution and pooling operations. This enables them to capture complex patterns and variations in images that are indicative of certain classes. In contrast, Random Forest, SVM, Naive Bayes, and Decision Tree are machine learning models that are designed to work with tabular data. They are less effective in dealing with high-dimensional data with spatial structure and may struggle to capture complex patterns and variations in images.

Therefore, in image classification tasks, CNNs often outperform other machine learning models in terms of F-score performance metric, as they are better able to capture the complex patterns and variations in images that are indicative of certain classes.

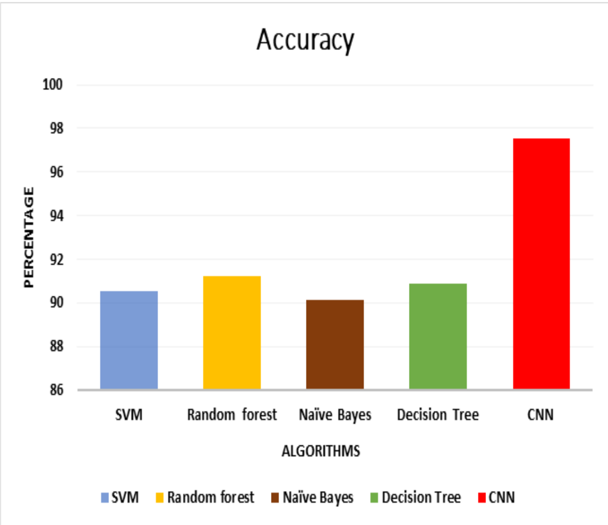

Figure 13. Performance Analysis for SVM, RF, NV, DT, and CNN

Figure 13 represents the presentation analysis of the chart model with compared deep learning algorithms, however this comparison analysis, the CNN model reached a better performance rate with an accuracy of 97.56% respectively.

CNNs are specifically designed for processing images and other high-dimensional data with spatial structure and are particularly well-suited for image classification tasks. Random Forest, SVM, Naive Bayes, and Decision Tree, on the other hand, are designed to work with tabular data and are more commonly used for tasks such as classification, regression, and clustering. In general, CNNs have been shown to achieve higher accuracy in image classification tasks compared to Random Forest, SVM, Naive Bayes, and Decision Tree. This is because CNNs can learn complex features and patterns in the image data, while the other models may struggle to capture the nuances of the images.

Decision trees and naïve Bayes are simpler and faster algorithms, but they may not perform as well as CNNs on complex image recognition tasks. SVMs can be effective for image classification, but they may require more tuning of hyperparameters and feature extraction compared to CNNs. Random forests are a popular ensemble method that can achieve high accuracy on a wide range of tasks, but they may require more training data and computational resources than CNNs.

In contrast, other machine learning algorithms may not capture all the relevant features of the data. Additionally, these algorithms may struggle with capturing spatial dependencies in images and handling non-linear relationships between input features, which can limit their accuracy for tasks like mathematical equation solving.

C. Performance Analysis of LeNet for different validation

OCR can be used for recognizing handwritten or printed characters in mathematical equations, which can then be used to build a parser to solve the equation. However, OCR may struggle with recognizing complex mathematical symbols or notation, and the accuracy of the OCR system can be influenced by the quality of the input image.

LeNet can be used for direct recognition of mathematical equations as images. This can be done by training the network on a large dataset of mathematical equations and then using the trained network to predict the solution to a new equation. This approach can achieve high accuracy and can handle complex mathematical symbols and notation.

The architecture of LeNet is simple, yet effective. It consists of multiple convolutional and pooling layers, followed by fully connected layers. This design helps LeNet to learn useful representations of the input images, which can then be used for recognition tasks. LeNet has been extensively tested on handwritten digit recognition tasks, and has been shown to achieve high accuracy rates. This suggests that the model is effective at recognizing patterns in images, which is a key component of OCR. LeNet is relatively efficient in terms of computational resources, meaning that it can be run on embedded schemes.

Overall, while there may be other models that perform well for OCR, LeNet is a strong contender due to its ability to effectively extract features from images and achieve high accuracy rates on image recognition tasks.

VGG is a deep CNN architecture that has shown high performance on a variety of image classification tasks, including recognizing handwritten digits[38][39]. VGG has a large number of layers and parameters, which can capture complex features and patterns in the input images. However, the large number of parameters can also make the network slower to train and more prone to overfitting. LeNet has fewer parameters and layers than VGG, which can make it faster to train and less prone to overfitting. However, the network may struggle to capture more complex features in the input images. LeNet was specifically designed for handwritten digit recognition, while VGG was designed for general-purpose image classification. This means that LeNet might perform better on tasks that are similar to its original design, such as recognizing other types of handwritten characters or symbols.

ResNet is a deep CNN architecture that has achieved state-of-the-art performance on various image recognition tasks[40]. However, it also has some drawbacks when compared to LeNet. ResNet requires a lot of computational resources to train and use. ResNet's deep architecture can also make it prone to overfitting, particularly when the dataset is small or noisy. This means that the network may perform well on the training data but poorly on new, unseen data. Handwritten mathematical symbols are often complex and have fine-grained details. LeNet uses smaller filter sizes and fewer pooling layers compared to ResNet, which can make it better suited for recognizing these details.

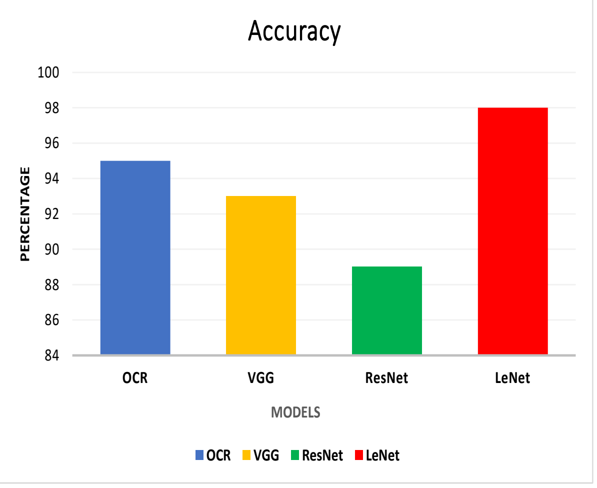

Figure14: Accuracy analysis of LeNet, OCR, VGG, and, ResNet

Figure 14 demonstrates that LeNet has the highest character recognition performance, with a score of 98%, followed by OCR at 95%, VGG at 93%, and ReNet at 89.02%.

V CONCLUSION

Using Convolutional Neural Networks (CNN), the handwritten mathematical equation solver is a fascinating and promising project that leverages deep learning to recognize handwritten mathematical symbols and equations. The project involves training a CNN model on a large dataset of handwritten mathematical symbols and equations to classify them accurately. The final system developed through this project can take a handwritten mathematical equation as input, recognize the individual symbols, and then use a mathematical parser to solve the equation. This system has the potential to be used in various applications, such as educational software, digital whiteboards, and handwriting recognition software.

Convolutional Neural Networks (CNN) is a powerful and effective machine learning technique for image classification and recognition tasks. Compared to decision trees, naïve Bayes, SVM, and random forests, CNNs have the advantage of automatically learning relevant features from raw input data, rather than relying on hand-engineered features. In summary, CNNs are a highly effective technique for image recognition tasks, especially when dealing with large datasets and complex images. LeNet is an effective CNN technique for the character recognitions task. It gives high accuracy compared with other CNN techniques such as OCR, ResNet, and VGG.

REFERENCE

1. PH.D, B. E. V. A. N. (2021). Artificial Intelligence and the Future of Humans. In Artificial Intelligence and the future of humans: Understanding the psychology, biology, and... challenges of ai. essay, PRAEGER PUBLISHERS.
2. Artificial Intelligence and its implications for the future of ETS's tests. (2014). Artificial Intelligence and the Future of Testing, 15–34. https://doi.org/10.4324/9781315808178-7
3. 21stCenturyTechnologies - OECD. (n.d.). Retrieved February 28, 2023, from https://www.oecd.org/futures/35391210.pdf
4. Ahmad-MontaserAwal, HaroldMouchère, Christian Viard-Gaudin.'Towards Handwritten Mathematical Expression Recognition' 2009 10th International Conference on Document Analysis and Recognition.
5. Guyon, I., & Wang, P. S.-P. (2014). In Advances in pattern recognition systems using neural network technologies: Principles and applications. essay, World Scientific Publishing Company.
6. Catherine Lu Karanveer Mohan.' Recognition of Online Handwritten Mathematical Expressions Using Convolutional Neural Networks' cs231n project report Stanford 2015.
7. Richard G. Casey, Eric Lecolinet " A survey of methods and strategies in character segmentation'1996
8. Iris segmentation and recognition using enhance dicentric segmentor... (n.d.). Retrieved February 25, 2023, from https://www.irjet.net/archives/V8/i12/IRJET-V8I12164.pdf
9. ZouaouiAbderaouf, BenblidiaNadjia, Oukid-KhouasSaliha "License plate character segmentation based on horizontal projection and connected component analysis" 2014 World

Symposium on Computer Applications & Research (WSCAR)

10. ZouaouiAbderaouf ‘license plate character segmentation based on horizontal projection and connected component analysis’ 2014 world symposium on computer applications & research (wscar).
11. Nebauer, C. (1998). Evaluation of convolutional neural networks for visual recognition. IEEE Transactions on Neural Networks, 9(4), 685–696. https://doi.org/10.1109/72.701181
12. Koonce, B. (2021). Convolutional neural networks with swift for tensorflow: Image recognition and dataset categorization. Apress.
13. Denise Schmandt-Besserat. Denise SchmandtBesserat. (n.d.). Retrieved March 12, 2023, from https://sites.utexas.edu/dsb/tokens/the-evolution-of-writing/
14. Blondel, V., & Megretski, A. (2004). Unsolved problems in mathematical systems and control theory. Princeton University Press.
15. Al-Jawfi, Rashad. "Handwriting Arabic character recognition LeNet using neural network." Int. Arab J. Inf. Technol. 6.3 (2009): 304-309.
16. Khanh Minh Phan, Cuong Tuan Nyugen, Anh Duc Le, Masaki Nakagawa, “An Incremental recognition method for online handwritten mathematical Expressions” , 3rd IAPR Asian conference on pattern recognition, 2015, 978-1-4799-6100-9, 171-175
17. Cuong Tuan Nguyen, Masaki Nakagawa, “ An improved segmentation of online English handwritten text using recurrent neural network” 3rd IAPR Asian conference on pattern recognition, 2015, 978-1-4799-6100-9, 176-180.
18. Mehryar Emambakhsh, Yulan He, Ian Nabney, “ Handwritten and machine – printed text discrimination using template matching approach” , IEEE, 12th IAPR workshop on document analysis systems, 2016, 978-1-5090-1792-8, 399-404.
19. Hubert Cecotti, "Deep Random Vector Functional Link Network for handwritten character recognition", Neural Networks (IJCNN) 2016 International Joint Conference on, pp. 3628-3633, 2016. ISBN ; 978-1-4799-1960-4. DOI : 10.1109/IJCNN.2015.7280451.
20. Liu, C., Lina Zuo, Xinfu Li, and Tian, X. (2015). “An improved algorithm for identifying mathematical formulas in the images of PDF documents”, 2015 IEEE International Conference on Progress in Informatics and Computing (PIC). doi:10.1109/pic.2015.7489848
21. Soni Chaturvedi, Rutika Titre, Neha Sondhiya, “Review of handwritten pattern recognition of digits and special characters using feed forward neural network and Izhikevich neural model ” , IEEE, International conference on electronic systems, signal processing and computing technologies, 9-11 Jan. 2014, ISBN: 978-1-4799-2102-7, 10.1109/ICESC.2014.83 208
22. Caiyun Ma, Hong Zhang, “Effective handwritten digit recognition based on multi-feature extraction and deep analysis” IEEE, 12th international conference on fuzzy system and knowledge discovery(FSKD), 2015, 978-1-4673-7682-2,297-301.
23. Rana S. Hussien, Azza A.Elkhidir, Mohamed G.Elnourani, “ Optical

character recognition of arabic handwritten characters using neural network ” , IEEE, International conference on computing , control, networking, electronics and embedded system engineering (ICCNEEE), 7-9 Sept. 2015, 978-1-4673-7869-7

24. Crohme: Competition on recognition of online handwritten mathematical expressions. TC11. (n.d.). Retrieved March 12, 2023, from http://www.iapr-tc11.org/mediawiki/index.php/CROHME:_Competition_on_Recognition_of_Online_Handwritten_Mathematical_Expressions
25. Sharma, P. (2023, February 6). *Image segmentation: Types of image segmentation*. Analytics Vidhya. Retrieved March 9, 2023, from https://www.analyticsvidhya.com/blog/2019/04/introduction-image-segmentation-techniques-python/
26. Vinod Nair, Geoffrey E. Hinton” Rectified Linear Units Improve Restricted Boltzmann Machines”2010
27. Leo Pauly , Peel, Shan Luo, David Hogg and Raul Fuentes.‘Deeper Networks for Pavement Crack Detection’34th International Symposium on Automation and Robotics in Construction (ISARC 2017)
28. Srivastava, N., G. Hinton, A. Krizhevsky, I. Sutskever. "Dropout: A Simple Way to Prevent Neural Networks from Overfitting." Journal of Machine Learning Research. Vol. 15, pp. 1929-1958, 2014.
29. Luo, Chunjie, et al. "Cosine normalization: Using cosine similarity instead of dot product in neural networks." Artificial Neural Networks and Machine Learning–ICANN 2018: 27th International Conference on Artificial Neural Networks, Rhodes, Greece, October 4-7, 2018, Proceedings, Part I 27. Springer International Publishing, 2018.Chakraborty, S., Chanda, B., & Das, A. K. (2018). Handwritten mathematical expression recognition: A survey on offline recognition. ACM Computing Surveys (CSUR), 51(3), 55.
30. Meurer, Aaron, et al. "SymPy: symbolic computing in Python." PeerJ Computer Science 3 (2017): e103.
31. Joyner, David, et al. "Open source computer algebra systems: SymPy." ACM Communications in Computer Algebra 45.3/4 (2012): 225-234.
32. S. Y. Chaganti, I. Nanda, K. R. Pandi, T. G. N. R. S. N. Prudhvith and N. Kumar, "Image Classification using SVM and CNN," 2020 International Conference on Computer Science, Engineering and Applications (ICCSEA), Gunupur, India, 2020, pp. 1-5, doi: 10.1109/ICCSEA49143.2020.9132851 .
33. Saragih, Glori Stephani, et al. "Ischemic stroke classification using random forests based on feature extraction of convolutional neural networks." International Journal on Advanced Science, Engineering and Information Technology 10.5 (2020): 2177-2182.
34. Mohapatra, Debaniranjan, Jogeswar Tripathy, and Tapas Kumar Patra. "Rice disease detection and monitoring using CNN and naive Bayes classification." Soft Computing Techniques and Applications: Proceeding of the International Conference on Computing and

Communication (IC3 2020). Springer Singapore, 2021.

35. Yin, Shouyi, et al. "Multi-CNN and decision tree based driving behavior evaluation." Proceedings of the Symposium on Applied Computing. 2017.
36. Priyam, Anuja, et al. "Comparative analysis of decision tree classification algorithms." International Journal of current engineering and technology 3.2 (2013): 334-337.
37. Zhang, Ye, and Byron Wallace. "A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification." arXiv preprint arXiv:1510.03820 (2015).
38. Shin, Hee-Ran, et al. "Performance improvement of optical character recognition for parts book using pre-processing of modified VGG model." The
39. Zhao, Haifeng, Yong Hu, and Jinxia Zhang. "Character recognition via a compact convolutional neural network." 2017 International conference on digital image computing: techniques and applications (DICTA). IEEE, 2017.0
40. Chen, Linhui, et al. "A modified inception-ResNet network with discriminant weighting loss for handwritten chinese character recognition." 2019 International Conference on Document Analysis and Recognition (ICDAR). IEEE, 2019.