www.ijecs.in
International Journal of Engineering and Computer Science
Volume 14 Issue 05 May 2025, Page No. 27183- 27193
ISSN: 2319-7242 DOI: 10.18535/ijecs/v14i05.5124

Letter Recognition from Noisy Images Using Deep Learning

Esraa Farouk Abou-Elmagd [1]*, Saddam Bekhet [1]
[1] Faculty of Commerce South Valley University Qena, Egypt.

Abstract

Machine learning and deep learning are being applied in multiple fields. Letter recognition presents a very active and challenging research area. Current research demonstrates a marked increase in focus on deep learning methodologies applied to letter recognition. Letter recognition is a problem that's been studied extensively in various languages. Letter recognition refers to the skill of recognizing and distinguishing letters based on their visual features. It includes examining response patterns and identifying particular elements that make up various letters. This perceptual process is essential for identifying distinct letters and is frequently examined through quick visual presentations and visual search activities. Furthermore, understanding letters in context explores how letters are perceived within words, thereby improving our understanding of reading processes. During the recognition process, pre-processing methods enhance image quality by minimizing noise and adjusting orientation, whereas convolutional neural networks are responsible for extracting features of the letters. The current letter recognition system encounters numerous difficulties in extracting text from noisy and distorted images or complex layouts, with extraction primarily restricted to numerical characters and the English alphabet. Many studies have employed deep learning models to enhance accuracy. The proposed method reaches an impressive 96.2% accuracy in identifying letters from input images.

Keywords: Deep Learning, Convolutional Neural Network (CNN), Letter Recognition, Noisy images.

1. Introduction

Pattern recognition is a crucial research field dedicated to identifying and interpreting different types of inputs, including images. Common applications of this field include letters, face, and speech recognition. A critical step in pattern recognition involves analyzing an object, identifying its key features, and comparing these with those of other objects to determine whether they match or differ. Letter recognition, a specific branch within pattern recognition, has gained significant research attention over the past few decades. Modern computers are capable of automatically recognizing and interpreting letters [1]. Devices can process letters from images and convert them into text format. Letter recognition refers to the process of identifying letters across different languages. This process typically involves two crucial phases: feature extraction and classification [2]. A typical letter recognition system usually follows four main stages: data collection, pre-processing, feature extraction, and classification: -

1. **Input Dataset:** To gather training and testing data, many researchers utilize publicly available standard datasets, while others develop and use custom datasets specific to their research needs.
2. **Pre-processing:** During this phase, several mathematical and morphological operations are applied to the input letter images. These include grayscale conversion, normalization, baseline detection, skew and slant detection and correction, noise removal, and image finalization.
3. **Feature Extraction:** Once the image is pre-processed, relevant features are extracted. These features are then

compared with those from the training dataset. If a match is found, the input image is identified accordingly.

4. **Classification:** In this final step, machine learning classifiers like Support Vector Machines (SVM) and Convolutional Neural Networks (CNN) are used to categorize the input based on the extracted features.

Letter recognition presents numerous challenges due to a variety of factors. One major difficulty stems from the diverse handwriting styles among individuals [3]. Additionally, the existence of different character types—such as uppercase and lowercase letters, as well as special symbols—adds further complexity to the task. To develop a highly accurate neural network model, a large and comprehensive dataset is essential. Another major obstacle is the presence of noise and distortions in input images, which can hinder accurate text detection. Although letter recognition is a vital area of machine learning, it remains a partially unsolved problem. Traditional neural networks have been employed to address it, but they face certain limitations. Consequently, deep learning techniques are being increasingly adopted to achieve improved accuracy and performance.

Figure 1 Examples of letters in Noisy Images

Numerous machines learning and deep learning techniques have been employed to tackle the problem of letter recognition [1]. Developing a practical and highly accurate letter recognition system remains a challenging task.

Developing a practical and highly accurate letter recognition system is still a difficult task. Deep Learning (DL), a modern approach within machine learning, focuses on learning data representations and has gained substantial prominence in object recognition due to its impressive performance improvements [4]. Today, deep learning is being applied across various domains, with letter recognition being a key area of interest. The application of deep learning (DL) has greatly improved the accuracy and overall performance of recognition systems [5].

DL methods are particularly effective in detection, classification, and learning tasks. These architectures extract features directly from raw data, removing the need for manual feature engineering and enabling more efficient and accurate model development. By leveraging multiple layers, DL systems extract meaningful patterns from raw input. Deep learning proves to be a powerful technique for feature extraction, especially in letter recognition tasks [2]. It enables the development of specialized models that learn hierarchical features from training data, building upon the foundations of traditional machine learning. In AI systems, DL techniques are widely adopted due to their efficiency and ability to maintain accuracy without escalating hardware costs. Their strength lies in learning from vast amounts of raw data, allowing the system to refine its feature extraction capabilities over time [6]. As a result, deep learning has garnered significant interest among researchers, particularly in solving complex computer vision problems such as detection, classification, and image recognition.

Among the most widely used DL algorithms is the Convolutional Neural Network (CNN) [7]. CNNs are advanced neural networks designed for computer vision tasks and are known for their ability to handle translation-invariant problems like object and speech recognition. They are especially effective for tasks such as visual image analysis, segmentation, and classification [1]. CNNs are comparatively easier and faster to train, with fewer parameters, and they tend to deliver reliable and accurate outcomes once properly trained. Their performance is closely tied to their capacity to process large volumes of raw input data [3]. Despite their high accuracy, CNNs can be computationally intensive due to their structural complexity. Letter recognition presents numerous challenges due to a variety of factors. One major difficulty stems from the diverse handwriting styles among individuals [3]. Additionally, the existence of different character types—such as uppercase and lowercase letters, as well as special symbols—adds further complexity to the task. To develop a highly accurate neural network model, a large and comprehensive dataset is essential. Another major obstacle is the presence of noise and distortions in input images, which can hinder accurate text detection. Although letter recognition is a critical area in machine learning, it is still a partially unsolved problem. Traditional neural networks have been employed to address it, but they face certain limitations. Consequently, deep learning techniques are being increasingly adopted to achieve improved accuracy and performance.

2. Literature Review

For nearly a century, psychologists have studied the human ability to recognize and produce letters [8]. Early studies focused on how people rapidly recognize machine-generated letter forms, which were often standardized in shape and size. Despite this uniformity, humans also demonstrate remarkable proficiency in identifying letters that appear in distorted or altered forms compared to standard printed text [9]. This adaptability in perception has led to the development of several theoretical models explaining how letter variants are processed by the brain. Among these are templates matching theories and those grounded in feature analysis [10].

Feature detection theories suggest that letter recognition involves identifying specific distinguishing features within each character [11]. One of the foundational models supporting this idea is Selfridge's Pandemonium Model of letter perception, introduced in 1959. According to this model, letters are recognized based on a set of defining features, with each feature triggering a response from metaphorical "demons" responsible for feature matching. The model also posits that the visual complexity of a letter—defined by the number of features it contains—directly impacts the recognition time, with more complex letters taking longer to identify [12].

Beyond the act of naming letters, another critical aspect of letter recognition is the ability to discern their defining features. This awareness may actually precede the ability to name or write letters and could significantly influence how quickly and accurately letter forms are learned and produced. Gibson et al. (1962) conducted foundational research on the critical features involved in letter recognition [13]. They observed that the difficulty in distinguishing letter features, such as horizontal lines, closed loops, or diagonal strokes, varied depending on transformations like orientation changes or added visual elements. This concept of "critical features" has influenced both letter recognition and formation. Building on this idea, According to Pelli and colleagues [14], participants could easily identify the seven unique visual features of letters, which include complexity, overlap, height, width, area, and visual efficiency. Recent studies have shifted toward the computational implementation of letter recognition. K. Swetha et al. (2021) conducted a comprehensive review of the primary challenges in constructing models that convert letter images into digital formats [15], while Ritik Dixit et al. (2021) examined the effectiveness of various algorithms—such as Support Vector Machines (SVM), Multi-Layer Perceptrons (MLP), and Convolutional Neural Networks (CNN)—for letter recognition tasks [16]. Their findings indicated that SVM achieved the highest training accuracy and the shortest execution time, whereas CNN excelled in testing accuracy despite longer processing durations. Fathma Siddique et al. (2020) further explored the impact of varying hidden layers and epochs on CNN performance using the MNIST dataset [17]. In another study, Ashish Shetty and Sanjeev Sharma (2023) emphasized the wide applicability of Optical Character Recognition (OCR) and highlighted the essential role of CNNs in pattern recognition tasks

[18]. Additionally, several researchers have investigated hybrid models to improve offline letter recognition. To improve recognition accuracy, one method integrated Artificial Neural Networks (ANNs) with Hidden Markov Models (HMMs) [20]. Other research [21-23] developed neural network-based systems that included pre-processing, data augmentation, convolutional and recurrent neural networks, and post-processing approaches. These models aimed to improve the accuracy and effectiveness of converting handwritten text into digital formats using MLPs and CNNs. Several recent studies have built upon earlier work to advance the field of letter recognition using deep learning. In [24], the authors introduced the first successful application of Convolutional Neural Networks (CNNs) to letter recognition, achieving state-of-the-art results on various benchmark datasets. In another study [25], researchers segmented approximately 90,000 images spanning over 40 different classes of the Devanagari script. They employed a deep learning architecture, leveraging CNNs for superior recognition performance compared to traditional shallow networks. Additionally, they implemented an "idler" mechanism, and a dataset increment approach to enhance testing accuracy. In [26], the authors addressed the same problem but adopted a holistic approach to recognize handwritten words, treating each word as a single unit. This method utilizes features such as density, long run, and structural characteristics extracted from handwritten document images. Classification was then performed using Support Vector Machines (SVMs). Another study [27] proposed a combined workflow and machine learning model aimed at recognizing letters in form images. This hybrid method uses CNNs for powerful feature extraction and SVMs as the final classifier. Experimental results showed that this hybrid approach outperformed CNNs alone, achieving higher accuracy. The model was further validated using ten-fold cross-validation, indicating room for further performance improvement. As computational power increases and resource constraints diminish, the field of letter recognition continues to evolve. Researchers face ongoing challenges in enhancing accuracy and robustness, calling for the development of new, innovative methods [28], there is a growing need for the development of novel methodologies:

1. **Letter Datasets:** Many existing letter recognition datasets lack sufficient size and diversity. There is a pressing need for comprehensive datasets that include a wide range of font styles, sizes, illumination conditions, user handwriting variations, and vocabulary. Expanding dataset diversity is essential for developing robust and general recognition systems.
2. **Noise Manipulation:** Unwanted noise in images can significantly degrade recognition accuracy, often resulting in misclassifications. While some studies have relied on manual noise removal, this approach is inefficient for large-scale datasets. Therefore, developing automated, scalable methods for noise detection and elimination remains a critical area of research.
3. **Low-Quality Documents:** Recognizing letters in degraded or ancient documents presents significant challenges because of the presence of noise, missing characters, and poor image quality. This remains an open research problem, requiring novel techniques for image enhancement and robust character extraction.
4. **Dataset Augmentation:** Data augmentation is a key strategy for enhancing model generalization. Overfitting can be reduced and recognition models' overall performance improved by artificially increasing the training dataset by transformations such as rotation, scaling, and flipping.
5. **Overfitting manipulation:** Overfitting remains a frequent challenge in deep learning models, where the system achieves high performance on training data but struggles to generalize to unseen data. Techniques such as data augmentation,

dropout, and regularization have proven effective in minimizing overfitting and improving model generalization.

In 2018, researchers in study [7] investigated the use of a Deep Neural Network (DNN) to classify letters from the EMNIST Letters dataset. The procedure started with extensive preprocessing steps, such as image segmentation, slant correction, character thinning using morphological operations, and image thresholding. Following preprocessing, the refined images were fed into the feature extraction and classification stages, which utilized DNN architecture. The DNN model employed a stacked autoencoder to facilitate layer-wise training. The network architecture featured two hidden layers containing 300 and 50 neurons, respectively, followed by a SoftMax output layer with 27 neurons representing each target letter class. This method attained a total classification accuracy of 88.8%.

In 2019, the study presented in [7] implemented the TextCaps model, which is based on a capsule network architecture. The input image undergoes processing through three layers of folding layers, that is, capsule or drawing capsule layers, which are fully connected to the primary capsule layer. Dynamic routing, executed over three iterations, links the primary capsules to the character capsules. When tested on the EMNIST Letters dataset, the model achieved an accuracy of 95.36% using the complete training dataset, and 92.79% accuracy when trained with only 200 samples per class.

A Deep Convolutional Neural Network (DCNN) model with Autonomous and Continuous Learning (ACL) capabilities has been evaluated in [7], enabling it to automatically design a DCNN architecture customized for a particular vision application. The optimizer used was RMSprop. The study utilized the VGG-5 architecture with a fully connected spinal cord structure, achieving an accuracy of 95.58%. When applied to the EMNIST Letters dataset, the best-performing model achieved an accuracy of 95.88%.

3. Background

Image processing refers to the application of various operations on an image to enhance its quality or to extract meaningful information from it [1]. It is a subset of signal processing where the input is an image, and the output can either be an improved version of the original image or specific features and characteristics derived from it.

3.1 The architecture of recognition process

1. **Pre-processing:** This is the initial phase of the recognition pipeline where raw image data is prepared for analysis [1]. It includes operations such as noise removal, slant correction, image fine-tuning, distortion elimination, and alignment. The goal of this stage is to improve image quality and ensure uniformity for subsequent processing.
2. **Feature extraction:** Once pre-processing is complete, relevant features are extracted from the cleaned image. This stage involves identifying patterns or characteristics (e.g., edges, curves, strokes) that are critical for distinguishing one letter from another [1]. These extracted features form the basis for the classification step.
3. **Classification:** In this phase, the extracted features are fed into a classification algorithm, which assigns the image to a specific category or label (e.g., a letter in the alphabet). This step is essential for enabling the model to learn and generalize from training data.
4. **Recognition:** The final step involves using a trained model to recognize and predict letters based on previously learned data. The processed and feature-extracted input is matched against the trained model to produce a recognition result, identifying the letter or symbol represented by the image.

3.2 Convolutional neural network

It can transform the input structure through each layer of the network, automatically extracting relevant features from the images [29].

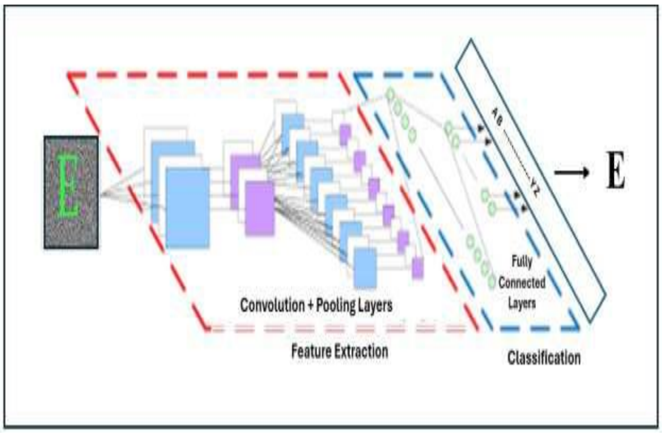


Figure 2 CNN Architecture

3.3 CNN Architecture

1. **Input Layer:** Acts as a buffer to hold the input data (e.g., an image) and passes it to the next layer for processing [30].
2. **Convolution layer:** This is the core layer responsible for feature extraction. It applies convolution operations on the input using a filter (kernel). As the kernel slides over the image (defined by the stride), it computes the sum of products at each location, producing feature maps. Multiple filters extract different features like edges, textures, and patterns [28].
3. **Batch Normalization layer:** Typically placed after the convolution layer and before the activation function, it normalizes the output using the batch's mean and variance. This helps to keep activations in a manageable range and speeds up training.
4. **RELU Layer:** Introduces non-linearity into the network. It performs an element-wise operation that replaces all negative values with zero, while keeping positive values unchanged. ReLU accelerates the convergence of the training process [31].
5. **Pooling Layer:** The spatial dimensions (width and height) of feature maps are reduced through pooling, which decreases computational load and helps prevent overfitting.
6. **Dropout layer:** To prevent overfitting, a subset of activations is randomly set to zero during training [30]. This encourages the model to learn more robust and general features.
7. **Fully connected layer:** Each neuron is interconnected with all of the neurons in the previous layer. This layer learns the high-level, non-linear combinations of features and is used for final decision making or classification [32].
8. **SoftMax Layer:** Usually appended to the end of a CNN in classification tasks It converts the fully connected layer's output into a probability distribution across all classes. Each value represents the network's confidence that a given input corresponds to a specific class [1].
9. **Classification Layer:** Computes the final loss—typically using the cross-entropy function—for mutually exclusive class labels. This layer helps the network evaluate its predictions against the ground truth and is essential for backpropagation during training.

4. Proposed Method

Deep Learning (DL) is a nonlinear computational technique used to learn patterns from data and predict complex trends, irrespective of the error distribution or the hidden intricacies within the data. It has been effectively applied across a wide range of domains, including Speech recognition, Facial recognition, Letter and handwriting recognition, Crop yield prediction and classification, Weather forecasting, Environmental monitoring, and Image fusion. Despite its success, a common challenge in neural network training is the tendency to get trapped in local minima or maxima, which can hinder optimal learning and performance. Among the various DL models, Convolutional Neural Networks (CNNs) stand out for their exceptional performance in image-related tasks. CNNs are particularly well-suited for image classification based on the ability to automatically and efficiently extract meaningful, high-level features that may not be easily interpretable by humans. CNNs work by Learning hierarchical patterns in visual data through multiple layers (convolutional, pooling, etc.). Extracting critical features such as edges, shapes, and textures. Classifying images based on the learned features. In the context of letter recognition, a CNN takes an image of a handwritten or printed letter as input, processes it through its layers, and outputs the predicted letter class. This automation enables high-speed and high-accuracy recognition, often without any human intervention.

4.1 Proposed Model Architecture

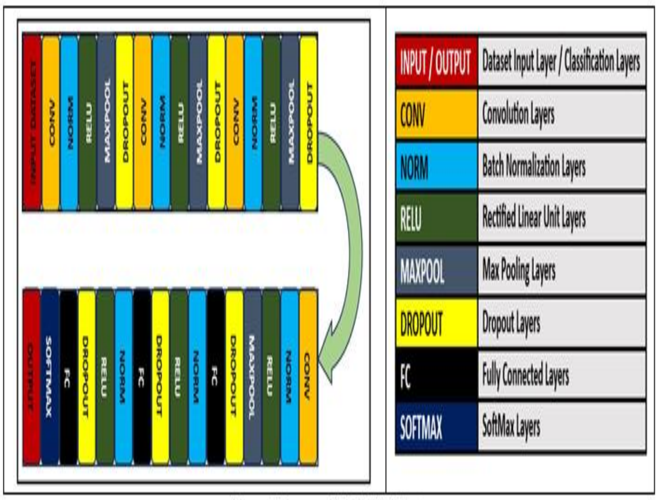


Figure 3 Proposed Model Architecture

In convolution layer consists of multiple filters that perform the process of convolution. Every image is represented as a matrix of pixel values, where each pixel holds specific information about the image. The way these matrices are structured depends on the type of image (grayscale or color). Then, Batch normalization helps to improve the stability and speed of training deep neural networks by normalizing activations across layers and mini batches, Solving the internal covariate shift problem, and allowing the use of higher learning rates. By ensuring the consistency of feature map distributions, BN also contributes to better convergence and can even reduce overfitting, making it an essential component in modern deep learning architectures. The ReLU operation applied elementwise to the feature map essentially removes the negative values and keeps the positive values, creating a rectified feature map that enhances the network's ability to learn complex patterns and reduces the computational cost due to sparsity. This rectified feature map is then passed to the next layer in the network for further processing. The pooling layer, particularly max pooling, reduces the spatial dimensions of the feature maps, enhancing the model's computational efficiency and making it more robust to minor variations in the input. By selecting the maximum value from each region, it captures the most prominent features, ensuring that important information is preserved while reducing the overall size of the data being processed. Dropout is a key regularization method in deep learning that helps prevent overfitting by deactivating neurons at random during training, ensuring that the model generalizes better to unseen data. By making the network less reliant on any particular neuron, dropout forces the model to learn distributed and robust representations of the data. Fully connected layers are essential for learning high-level patterns and solving the internal covariate shift problem

king classification decisions. However, dropout and regularization are required to avoid overfitting because of the large number of parameters. The output from the FC layer is a feature vector, which is used to predict the class label for the input, usually through a SoftMax layer for multi-class classification tasks. SoftMax is used as the final activation layer in a CNN for classification tasks. It converts the raw output of the network (logits) into a probability distribution, with each class having a probability between 0 and 1. This ensures that the network's prediction is both interpretable and meaningful, with the class corresponding to the highest probability being considered as the final output. Finally, classification of the given input into distinct classes takes place at this layer.

5. Experiments and Results

5.1 Dataset

The dataset has 82 image categories. The images represent the English alphabet (A-Z), digits, and some symbols. It can be a useful resource for researchers, developers, and students working on projects related to computer vision, machine learning, and deep learning.

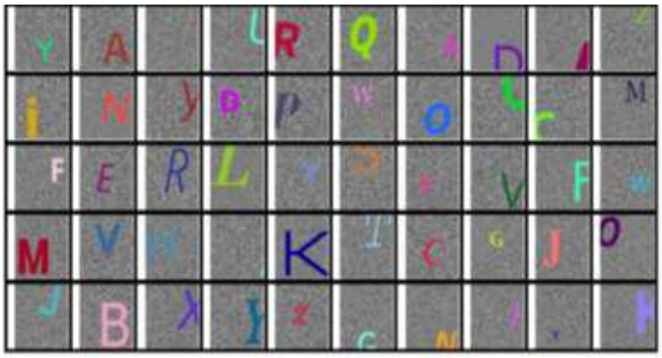


Figure 4 samples of dataset

5.2 Deep Learning

Deep learning effectively extracts relevant features that help differentiate between 26 letters, 10 digits, and other symbols. For the first time, the dataset previously used for the letter recognition problem is the input to the deep learning phase. This dataset consists of a collection of labeled images that include problem categories. Once the dataset is passed to the input layer of a convolutional neural network (CNN), it goes through multiple convolution, activation, and pooling layers to extract important features. A collection of labeled images is the output of the last layer (the fully connected layer), each associated with a probability indicating the model's confidence for each class. The class with the highest probability is selected as the predicted label for the given input.

6. Performance of Technique

The Proposed CNN Network is experimented with using other networks.

6.1 Accuracy of Deep Learning Stage

In accordance with a standard experimental design, 70% of the images in each dataset were assigned to training and 30% to validation at random.

Additionally, the accuracy measure, which computes the percentage of successfully recognized images to the total number of images in the dataset, is used to evaluate the model's performance. This is a straightforward metric that indicates how often the model's predictions match the true labels.

Mathematically, accuracy is expressed as:

$$\textbf{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

Furthermore, the accuracy measure is used to evaluate the model's performance by calculating the percentage of successfully recognized images compared to the total number of images in the dataset. It is a straightforward metric that shows how often the model's predictions match the true labels.

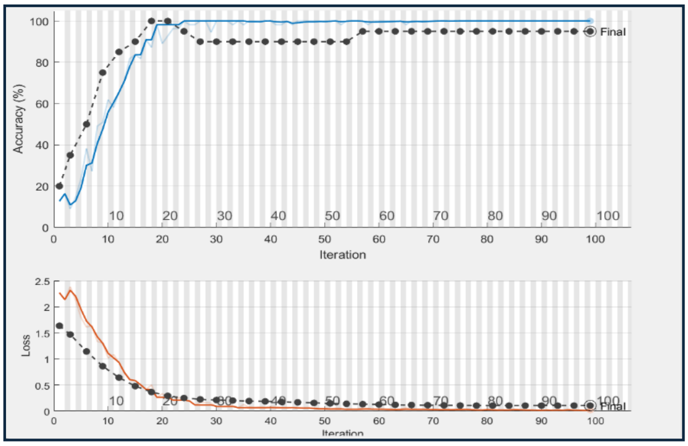
6.2 Biometric Metric for (CNNGA) Model

The accuracy measure, which assesses the percentage of successfully recognized images in the total dataset, is used to evaluate the research's performance.

Method	Accuracy	FRR	FAR	EER
Proposed Model	96.2%	4.9%	5.2%	5.5%

Another performance measure is biometric metrics, which are used to evaluate the effectiveness of the proposed technique in systems that involve biometric recognition. These metrics help assess the performance of the system in terms of its ability to correctly classify legitimate inputs (e.g., valid users or correct data) and reject invalid ones. Some important biometric metrics include: FRR measures the percentage of legitimate inputs (valid users or correct data) that the system incorrectly rejects. A higher FRR indicates that valid inputs are being erroneously rejected, which can degrade the user experience or the system's accuracy. FAR quantifies the percentage of invalid inputs (incorrect users or fake data) that the system mistakenly accepts as valid. A higher FAR suggests that the system is too lenient in accepting data, which can compromise security or accuracy. EER is the point at which the FRR and

FAR are equal. It provides a balanced measure of the system's accuracy, showing when the rates of false rejections and false acceptances are at the same level. A lower EER is desirable as it indicates better performance, with fewer mistakes in both rejection and acceptance of data.



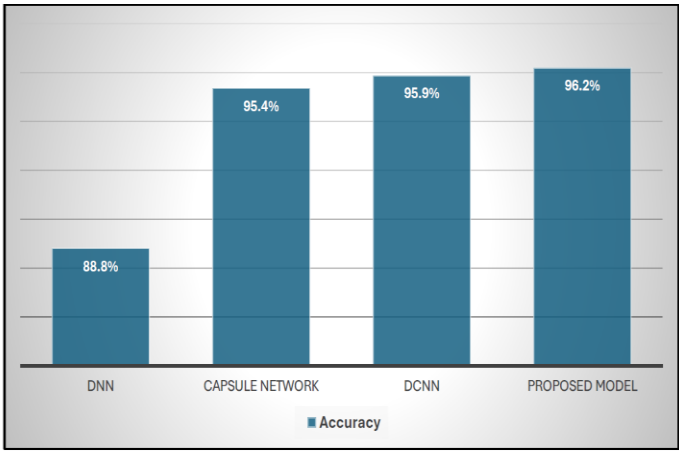


Figure 5 Comparison between proposed models and other models

7. Conclusion

In this paper, an efficient deep learning method for letter recognition from input photos was presented. The proposed CNN model achieved an impressive accuracy of 96.2%, despite the challenges posed by noisy and cropped input images. To address these challenges and improve the model's robustness, additional layers were incorporated into the deep learning architecture. These layers help enhance the model's ability to recognize and classify letters, even when the images are distorted or partially cropped.

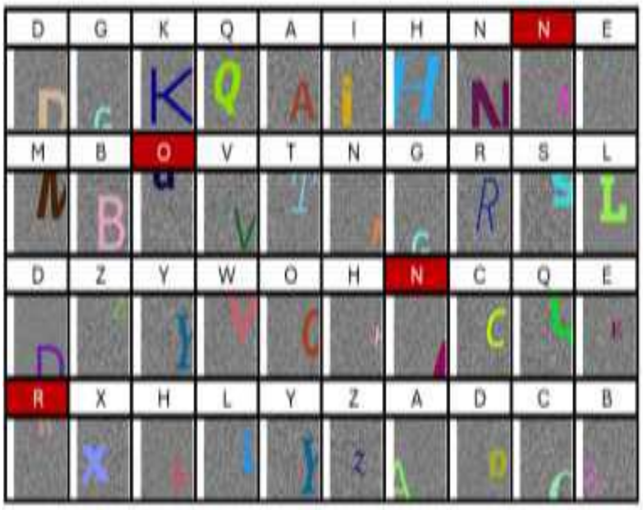


Figure 6 Samples of proposed model output

Future Work

Future work will focus on increasing the OCR model's accuracy by gathering more diverse and representative training datasets, which can help the model generalize better to a wider range of input variations. Another direction for future work is enabling OCR systems to recognize letters across multiple languages, expanding their versatility and real-world applicability.

References

1. Srivastava, Ankur, et al. "Handwritten character recognition using machine learning and deep learning." Emerging Trends in IoT and Computing Technologies. Routledge, 2023. 475-482.
2. Bhatt, Preeti P., and Isha Patel. "Optical character recognition using deep learning-a technical review." National Journal of System and Information Technology 11.1 (2018): 55.
3. Saqib, Nazmus, et al. "Convolutional-neural-network-based handwritten character recognition: an approach with massive multisource data." Algorithms 15.4 (2022): 129.
4. Younis, Khaled S. "Arabic hand-written character recognition based on deep convolutional neural networks." Jordanian Journal of Computers and Information Technology 3.3 (2017).

5. Balaha, Hossam Magdy, et al. "A new Arabic handwritten character recognition deep learning system (AHCR-DLS)." Neural Computing and Applications 33 (2021): 6325-6367.
6. Munappy, Aiswarya Raj, et al. "Data management for production quality deep learning models: Challenges and solutions." Journal of Systems and Software 191 (2022): 111359.
7. Liman, Muhamad Arief, Antonio Josef, and Gede Putra Kusuma. "Handwritten character recognition using deep learning algorithm with machine learning classifier." JOIV: International Journal on Informatics Visualization 8.1 (2024): 150-157.
8. Reutzel, Pamela, Kathleen AJ Mohr, and Cindy D. Jones. "Exploring the relationship between letter recognition and handwriting in early literacy development." Journal of Early Childhood Literacy 19.3 (2019): 349-374.
9. Neisser, Ulric, and Paul Weene. "A note on human recognition of hand-printed characters." Information and Control 3.2 (1960): 191-196.
10. Naus, Mary J., and Robert J. Shillman. "Why a Y is not a V: a new look at the distinctive features of letters." Journal of Experimental Psychology: Human Perception and Performance 2.3 (1976): 394.
11. Neisser, Ulric, and Henry K. Beller. "Searching through word lists." British Journal of Psychology 56.4 (1965): 349-358.
12. Pelli, Denis G., Bart Farell, and Deborah C. Moore. "The remarkable inefficiency of word recognition." Nature 423.6941 (2003): 752-756.
13. Gibson, Eleanor P., et al. "A developmental study of the discrimination of letter-like forms." Journal of comparative and physiological psychology 55.6 (1962): 897.
14. Pelli, Denis G., et al. "Identifying letters." Vision Research 46.28 (2006): 4646-4674.
15. Sethi, Rohan, and Ila Kaushik. "Handwritten digit recognition using machine learning." 2020 IEEE 9th International Conference on Communication Systems and Network Technologies (CSNT). IEEE, 2020.
16. Pashine, Samay, Ritik Dixit, and Rishika Kushwah. "Handwritten digit recognition using machine and deep learning algorithms." arXiv preprint arXiv:2106.12614 (2021).
17. Rai, Ankur, Siddharth Singh, and Sumit Kushwaha. "Optimized Handwritten Digit Recognition: A Convolutional Neural Network Approach." 2024 International Conference on Communication, Control, and Intelligent Systems (CCIS). IEEE, 2024.
18. Shetty, Ashish, and Sanjeev Sharma. "Ensemble deep learning model for optical character recognition." Multimedia Tools and Applications 83.4 (2024): 11411-11431.
19. Rakesh, S., et al. "Handwritten text recognition using deep learning techniques: A survey." MATEC Web of Conferences. Vol. 392. EDP Sciences, 2024.
20. Graves, Alex, et al. "A novel connectionist system for unconstrained handwriting recognition." IEEE transactions on pattern analysis and machine intelligence 31.5 (2008): 855-868.
21. Nikitha, A., J. Geetha, and D. S. JayaLakshmi. "Handwritten text recognition using deep learning." 2020 International Conference on Recent Trends on Electronics, Information, Communication & Technology (RTEICT). IEEE, 2020.
22. Graves, Alex, and Jürgen Schmidhuber. "Offline handwriting recognition with multidimensional recurrent neural networks." Advances in neural information processing systems 21 (2008).

23. Liwicki, Marcus, et al. "A novel approach to on-line handwriting recognition based on bidirectional long short-term memory networks." Proceedings of the 9th International Conference on Document Analysis and Recognition, ICDAR 2007. 2007.
24. LeCun, Yann, et al. "Gradient-based learning applied to document recognition." Proceedings of the IEEE 86.11 (1998): 2278-2324.
25. Arora, Sandhya, et al. "Devanagari Character Recognition: A Comprehensive Literature Review." IEEE Access (2024).
26. Madhvanath, Sriganesh, and Venu Govindaraju. "The role of holistic paradigms in handwritten word recognition." IEEE transactions on pattern analysis and machine intelligence 23.2 (2001): 149-164.
27. Fanany, Mohamad Ivan. "Handwriting recognition on form document using convolutional neural network and support vector machines (CNN-SVM)." 2017 5th international conference on information and communication technology (ICoIC7). IEEE, 2017.
28. Balaha, Hossam Magdy, et al. "A new Arabic handwritten character recognition deep learning system (AHCR-DLS)." Neural Computing and Applications 33 (2021): 6325-6367.
29. Younis, Khaled S. "Arabic hand-written character recognition based on deep convolutional neural networks." Jordanian Journal of Computers and Information Technology 3.3 (2017).
30. Bora, Mayur Bhargab, et al. "Handwritten character recognition from images using CNN-ECOC." Procedia Computer Science 167 (2020): 2403-2409.
31. Lahreche, Afafe, and Abdelhakim Cheriet. "Deep learning for Arabic letters recognition." (2019).

Author Profile

Esraa Farouk received the M.Sc. degree and PhD in Computer Science from Assiut University, Assiut, Egypt. She currently works as a professor at Department of Quantitative Methods, Faculty of Commerce, South Valley University, Qena, Egypt. Email: esraa.mohamed11@com.svu.edu.eg

Saddam Bekhet received the M.Sc. degree in Computer Science from Assiut University, Assiut, Egypt, in 2010 and a Ph.D. degree in Computer Science from University of Lincoln, UK, in 2016. He currently works as Associate professor at Department of Quantitative Methods, Faculty of Commerce, South Valley University, Egypt. His current research interests include content-based video retrieval, especially from the visual content perspective, video and scene understanding, human identification and deep learning. Email: saddam.bekhet@svu.edu.eg