Communications on Applied Nonlinear Analysis
ISSN: 1074-133X
Vol 32 No. ICMASD (2025)

Comparative Study of Character Recognition for Handwritten Characters

Tarun Kumar[1] , Arvin Vinayek[1] , Pradip Kumar Yadava[2]

[1]School of Engineering and Technology, CT University, Ludhiana, Punjab, 142024, India

[2] Department of Computer Science , Sunrise University Alwar , Rajasthan, 301028 , India

Article History:

Received: 12-12-2024

Revised: 25-01-2025

Accepted: 05-02-2025

Abstract:

Optical Character Recognition (OCR) represents an important technology in the context of computer vision, which facilitates the extraction of textual data with the help images. Although a lot of investigations have examined various OCR models, there is a notable absence of comparative studies evaluating different algorithms on a unified standardized dataset. This research targets to fill this void by assessing multiple OCR models across two distinct datasets: one consisting of 28×28-pixel images and the other comprising of 64×64-pixel images. The models evaluated include ten Convolutional Neural Networks (CNNs) characterized by diverse activation functions, architectural depths, and, dropout rates, in addition to Long Short-Term Memory (LSTM) networks, Support Vector Machines (SVM), Encoder-Decoder frameworks, and Random Forest classifiers.

Through our analysis, it was revealed that the CNN-based models demonstrate exceptional performance, with the leading 64×64 CNN model achieving an accuracy of 0.9882, while the highest-performing 28×28 CNN model reported an accuracy of 0.9763. The Encoder-Decoder model also showed formidable results, achieving an accuracy of 0.9781 on the 64×64 dataset and 0.9810 on the 28×28 dataset. SVM exhibited robust performance on the higher-resolution dataset, achieving an accuracy of 0.9777, but encountered significant challenges on the 28×28 dataset, where the accuracy of only 0.7252 was reported. The Random Forest classifier maintained a consistent accuracy of 0.9538 across both datasets. Conversely, LSTM models struggled to generalize effectively for OCR applications, with the best LSTM model achieving a mere 0.0346 accuracy on the 64×64 dataset and performing poorly on the 28×28 dataset as well.

Through the comprehensive study of CNN model, it was observed that, Model 3 attained the highest accuracy of 0.9907 across both datasets, accompanied by minimal validation loss (0.0377 for 28×28 and 0.0577 for 64×64). Other CNN models exhibited varying levels of performance, with deeper architectures generally surpassing their shallower counterparts. Our methodology encompassed preprocessing the datasets, partitioning them into training and testing sets, and training each model with suitable hyperparameters. The findings underscore that CNN-based architectures are the most effective for OCR tasks, particularly at elevated resolutions.

These results offer significant insights into the efficacy of various OCR models.

Keywords: Recognition, comprehensive, encompassed, hyperparameters.

Introduction

Optical Character Recognition (OCR) powered by Machine learning has become an important tool across various industries and applications. The ability to convert printed or handwritten text into a digital format has made it vital for tasks such as document digitization, automated verification, and real-time translation. By utilizing machine learning algorithms, OCR systems improve their text recognition and processing capabilities, enhancing efficiency and reducing the manual effort needed to handle large volumes of textual data. This technology is widely used in areas like document scanning, license plate recognition, automated data entry, and accessibility solutions for individuals with visual impairments.

OCR can primarily be categorized into two types: offline and online recognition. Offline OCR is used to identify printed or handwritten text from scanned documents, books, or images, analyzing the input after it has been captured. On the other hand, online OCR focuses on real-time text recognition, often involving digital writing inputs, such as handwriting with a stylus on touch-sensitive screens. Both types rely on advanced preprocessing techniques to enhance text recognition accuracy by reducing noise, normalizing inputs, and segmenting characters. The effectiveness of OCR is affected by several factors, which include, the quality of input images, feature extraction methods, classification models, and the specific language or script being processed.

Regardless of significant improvements in OCR technology, there are some challenges. Variations in handwriting styles, low-quality scanned documents, distorted or overlapping text, and complex character structures pose significant hurdles to achieve high accuracy. While deep learning-based OCR models have shown ample improvements, traditional methods still hold relevance in situations where computational efficiency is critical. The growing demand for multilingual OCR systems further complicates the landscape.

The main objectives of this research are:

- To summarize existing OCR research based on different languages and machine learning techniques.
- To highlight the strengths and weaknesses of various OCR models.
- To identify the most efficient machine learning-based OCR model for standardized datasets.
- To provide a comparative analysis of different OCR techniques based on training and testing accuracy.

Research in the field of Optical Character Recognition (OCR) has investigated a diverse array of methodologies, encompassing both conventional machine learning techniques, such as Support Vector Machines and Hidden Markov Models, as well as more sophisticated deep learning strategies, including Convolutional Neural Networks and Long Short-Term Memory networks. Additionally, hybrid models that integrate multiple methodologies have been developed to improve performance, especially in the context of recognizing handwritten or degraded text. Despite numerous studies reporting high accuracy for specific models, there is a scarcity of direct comparisons among different techniques, mainly because of discrepancies in datasets and evaluation criteria.

The methodology employed in this research adopts a structured framework for evaluating various OCR models. This process includes data preprocessing, the selection of different machine learning and deep learning models, training these models on varying image resolutions, and evaluating their performance. The effectiveness of each model is quantified through metrics such as accuracy, precision, and computational efficiency, with the goal of identifying the most effective approach. By conducting comparisons of different models under uniform conditions, this research seeks to elucidate their respective strengths and weaknesses.

The outcomes of this study reveal that OCR models based on deep learning, particularly those utilizing CNN architectures, surpass traditional methods in the recognition of both printed and handwritten text. Enhanced image resolution correlates with improved recognition accuracy, although certain models exhibit difficulties when faced with variations in handwriting and distorted inputs. These findings underscore the critical role of selecting suitable preprocessing techniques and model architectures to achieve optimal performance in OCR tasks.

Although considerable advancements have been made in optical character recognition (OCR) technology, several challenges persist, including the need to enhance accuracy for low-quality inputs, accommodate a variety of handwriting styles, and improve computational efficiency. Future research is anticipated to concentrate on the integration of OCR with natural language processing to achieve superior contextual comprehension, the creation of lightweight models suitable for real-time applications, and the enhancement of multilingual OCR functionalities. Given the increasing demand for automation and digital text processing, OCR continues to be an essential technology with significant implications across multiple industries and everyday uses.

Literature Review

This paper aims to examine current models and machine learning methods used for optical character recognition. Different languages use different feature extraction methods and classify with different classification methods.

Raus et al. [1] used ANN to recognize the numeral character on the licence plate. The images are compared pixel by pixel to get the information of difference between two images and characters. The recognition rate comes out to be 98.2%. Here each pixel act as input neuron for artificial neural network. Connell et al. 2000 proposed to recognise Hindi character written on screen through pen. A Devanagari character experiment with more than 20 different writers and each giving 5 datasets is conducted which resulted in 86.5% recognition rate. Here both Hidden Markov model (for online points) and NN for offline images are used for recognition.[2]

N. Arica et al. 2001 [3] talks about the steps taken by the character recognition process and their different types and variations. [4] Ramakrishnan n.d. 2002 created model which supports both Tamil and Roman scripts. The accuracy for Tamil, English in SVM is 88.43 and 87.76% while in case of Neural network it is 73.61 and 71.23% and for K-NN it I 68.25 and 84.72% respectively. Ashwin et al. 2002 [5] provide a fresh set of computationally straightforward features for the recognition problem. To obtain the final recognition, several 2-class classifiers were used utilizing the Support Vector Machine (SVM) technique. Arica et al. 2002 [6] provided a novel analytical framework for the offline recognition of cursive handwriting. Slant angle, stock height and width, baseline are used for feature

extraction. Third, character candidates are labelled and ranked using Hidden Markov Models (HMM) for shape identification. Different word sizes have different recognition rates in the entire testing space. This results in 88.3% accuracy with 30,000 words.

Bajaj et al. 2002 [7] proposed to classify Devanagari numerical through multi-classifier that have different type of features to classify. This paper uses Density feature, Moment feature of 4 directional curves and Descriptive component feature to abstract the information about the image. A 3-layer multi-layer perceptron is used with KNN at its base layer with back propagation for classification. The model is tested with 2460 samples and accuracy is 89.68 percent. Cowell et al. 2003 [8] highlighted the challenges of using standard structural and syntactic recognition algorithms when describing the Amharic script. First the character is measured against several templates and compares it to a set of template signatures. 377 pairings have ratings of 90 or above. In addition, two pairs have ratings of 99 and 23 pairs have ratings of 98. Noor et al. 2004 [9] address the issue of character rotation while recognition of English characters. With more than 76% accuracy for both classifiers in the chosen test set. It results in more than 90% accuracy with some characters showing low accuracy while rotated like “H”, “M”, “W” etc. Fukumoto et al. 2004 [10] focuses on improving the GLVQ's ability to recognize handwritten characters with greater precision (generalized learning vector quantization). FDA is combined with GLVQ in previous paper to identify the handwritten Chinese character using Euclidean distance which are gated and evaluated to be very effective. The accuracy increases to 99.6% for GVQ, FD and projectile distance used all together. Sharma et al. 2006 [11] proposed a quadric classifier for offline Devanagari character recognition. Direction chain code is used which gives 98.86% and 80.36% accuracy for numerals and characters respectively. Hanmandlu et al. 2007 [12] recognized Hindi characters using normalized distances as feature and exponential function fitted fuzzy network to classify the characters. Reuse policy is used under reinforcement learning which increases accuracy by 25 times when learning. 4750 samples give an accuracy of 90.65%. Pal et al. 2007 [13] tries to recognize the numeric in Indic version using quadric classifier. The images are first passed through directional chain code which then goes through gaussian filter which result in the features extracted which result in 99.56% accuracy. Pal et al. 2008 [14] proposed to increase the accuracy of Devanagari character by using directional feature and curvature feature and for classification SVM and MQDF are used with 36172 dataset and got 95.13% accuracy. We saw that different feature extraction techniques are used for better accuracy.

Mali et al. 2010 [15] proposed to classify individual Marathi numerals written in Devanagari script where 64 Fourier descriptor are used to describe the shape of characters. 3 classifiers NN, KNN and SVM are used as classification method which result in 97.05%, 97.04%and 97.85% with 13000 samples. Pal et al. n.d. 2010 [16] used a multilayer perceptron with a single hidden layer to identify handwritten English characters. Boundary tracing and the Fourier Descriptor are the features taken from the handwritten character.500 samples were used for the test with a backpropagation network can recognize handwritten English letters with an accuracy of 94%.Nasien et al. 2010 proposes using the "Freeman chain code (FCC)" to represent an image character in a recognition model for recognizing English handwritten characters. Chain codes give a way to define the bounds of a character picture by indicating the location in which the preceding pixel will be positioned. SVM, or support vector machines, are used for the classification phase. The experiment uses data from MNIST databases. The

SVM model is established using the radial basis function. The links between the parameters of kernel are finally discovered to improve training and testing accuracy. It shows 86% , 88% , 73% accuracy for first , second and third set of datasets.[17]

Yadav et al. 2013 worked on Hindi language as it is very difficult than English as it does not have character separation. An ANN algorithm is used for Hindi character recognition using the grayscale images processed to same level or normalized. A back-propagation neural network with two hidden layers is utilized to create the neural classifier. In order to build the neural classifiers, a gradient decent based neural network with two hidden layers is used. The model has been trained and evaluated on Hindi texts from print. A classification results of 90% is possible in performing.[18] Gaur et al. 2015 a three-stage process is used to recognize Hindi characters. The classification procedure is the third stage, and support vector machines are employed for this. Both Euclidean distance and a SVM are used to classify the results in this calculation. SVM produces better outcomes than Euclidean distance does. The highest outcome obtained using Euclidean distance is 81.7%. SVM is employed with a linear kernel, and the result is 95.86%. [19] Sonawane et al. 2015 compared the accuracy of two OCRs for English language. It is a component of ANN and the "nearest neighbor" method are the two recognizers that are employed. Binary digits must be used to represent the characters. To do this, the symbols were combined in a matrix of 7 columns and 5 rows. The implementers of the recognizers on the extracted features were done using a math-lab program. The outcome demonstrates that when used with English characters, Nearest Neighbor is a better character recognizer. Accuracy of NN is nearly 57.69% and of Nearest Neighbor method as 61.53%. The outcome demonstrates unequivocally that the "Nearest Neighbor" Approach outperforms.[20]

Singh 2018 used piecewise Histogram of oriented gradient to collect information from partitioned images to recognize Devanagari language thus this paper uses a piecewise feature extraction method. To classify a neural network is used which is trained with partitioned HOG information. It acquires the maximum accuracy of 99.27% and average of 97.06% of accuracy.[21] Lamghari et al. 2018 presented an offline character recognition process for Arabic handwritten characters. Here five methods are used to extract 66 structural, reginal and statical characteristics which are treated in a feed forward neural network with hidden layers. Database for Arabic handwritten characters and ligature (DAHCL) is used for training and testing which results in 98.27% accuracy. [22] Jebril proposes a character recognition system for Arabic language. It uses HOG for feature abstraction and SVM for classification. Paper uses its own dataset of 43000 where 30000 is used for training purposes and 13000 for testing purposes. The result tends to be successful as it provides an accuracy of above 99%.[23]

Koellner et al. [24] 2019 proposed a methodology for online character recognition. Here input is taken by gyro meter and the time instance, acceleration in different directions which then is changed into information through Fast Fourier Transform, Wavelet and Autocorrelation functions. This information is used with KNN, Linear Discriminant Analysis and Naïve Bayes as classification phase. Nearly a dataset of 20,000 lower case alphabet is used provided by 15 people. The result comes out to be different depending on if dataset used for training is of the user or not. For user dependent dataset and training accuracy is 95% , 92% and 84% while for general dataset it is 36% , 46% and 52% for KNN , LDA and LSTM respectively. Gurbuz et al. 2020 used RFF sensor for sign language where it uses simple machine learning technology to achieve accuracy of 72.5%.[25]

Liu et al. 2020 proposed a novel idea of a model called stroke sequence-dependent deep convolutional neural network (SSDCNN) which uses the eight directional method and stroke sequence to create a Chinese character recognition system. The process is mainly divided into two parts 1. Creating the stack of strokes in the writing order of them 2. Representation of character in terms of directional features. After completing the above two steps both features are combined with each other, and neural network is trained. This model has high accuracy of 97.86% while the error rate is 58.28% lower than the eight directions feature alone model.[26]

Saez-Mingorance et al. 2021 [27] uses ultrasonic transceiver with CNN, LSTM , convolutional autoencoder for character recognition. As result it gets 99.51% accuracy with 71 ms latency.

Babiker Hamdan 2021 compares different feature extraction methods and classification methods to recognize English character and numerals. This paper tries to improve accuracy with the help of combination of different techniques. Statical method, Template matching method, Structural pattern acknowledgement and statical SVM are used which give accuracy of 86 , 65 , 75 , 88 percent and sensitivity is 82 , 70 , 71 , 89 percentage respectively.[28]

Alemayoh et al. 2022 [29] develops a deep learning smart compact pen that can recognize 36 letters. It uses DNN, LSTM , CNN to create a network that has validation accuracy of 99.05%. Zhang et al. 2022 proposes a device to recognize the letter written by finger when worn by it. This device uses deep learning like CNN to recognize the 36 characters (26 English + 10 Numerals) which results in 97.95% accuracy.[30]

Bhattacharyya et al. 2022 proposed a two-stage deep feature selection approach for recognition of Devanagari and Bangla language. First the features are ranked by filter method RelieF and then the ranked feature are optimized by gray wolf. As result accuracy reaches 100% for Bangla and 99.61% for Devanagari. [31]

Zhang et al. 2023 proposes a finger writing recognition using time of Flight distance tool to take input. The database of 21 users is taken for English language. Deep learning algorithms such as LSTM, CNN are used for classification where LSTM gives best result with 98.31% accuracy.[32]

Bwatiramba et al. 2023 uses CNN and SNN for Numerals character recognition. It achieves 98.13% accuracy. [33]Bhatti et al. 2023 proposes to recognize Urdu language using CNN for feature extraction and SVM as classifier. It gives an accuracy of 98.41% for CNN only and its 99% for both CNN and SVM.[34] Darshni et al. 2023 develops a character recognition system with help of sciLab using ANN to classify the classes. The character used dare numerals thus total classes to be recognized are divided in 10. The classification is proceeded by online backpropagation with help of changing weights. A 2-layer ANN is used which results in a lower error rate and accuracy of 99.92%. With standard backpropagation the accuracy came to be 99.62%. [35]

From all the above methods and models, we get to know that different models and machine learning techniques are used for precise character recognition. Some advanced techniques like CNN alone can be used for character recognition with high accuracy but time consumption for its training and prediction is very high. In fast moving world these techniques remain behind the combinations of different feature extraction and classification methods which provide very little complexity and time

consumption for high accuracy. Table 1 shows different languages , techniques used for feature extraction and classification model.

Table 1: Comparison Table

Ref. No.	Language	Feature Extraction	Classification	Result(%)
[1]	Numeral	Direct Comparison	ANN	98.2
[35]	Numeral	ANN	ANN	99.92
[31]	Devanagari , Bangla	RelieF + Gray wolf optimizer	CNN	99.61,100 (nearly)
[33]	Devanagari	CNN	CNN	98.13
[17]	English	Freeman chain code	SVM	88
[23]	Arabic	HOG	SVM	99
[28]	English + Numeral	Struct. + Temporal pattern	SVM	88
[7]	Devnagari	Density + Moment feature	KNN	89.68
[16]	English	Boundary tracer	Gradient Decent NN	94
[18]	Hindi	NN	Gradient Decent NN	90
[6]	Cursive	Global parameters	HMM	88.3
[26]	Chinese	Eight Directional	SSDCNN	97.86
[20]	English		ANN,NN	57.69 , 61.53
[34]	Urdu	CNN	CNN + SVM	99
[22]	Arabic	Structural characteristics	FFD-NN	98.27
[9]	English	Geometric features	Fuzzy Logic	90
[10]	Chinese	Euclidean Distance	GLVQ + FDA	99.6
[2]	Hindi		HMM +NN	86.5
[24]	English	FFT + Wavelet	KNN , LDA , LSTM	95,92,84
[19]	Hindi	Euclidean Distance	KNN + K-means	95.86
[32]	English	CNN	LSTM + CNN	98.31
[21]	Devanagari	Piecewise HOG	ANN	99.27
[15]	Numeral	Fourier descriptor	ANN	97
[13]	Indic Numeral	Directional chain code	Quadratic classifier	99.56
[3]	Tamil , English		SVM NN KNN	88.4 ,87.7 73.6 ,71.2 68.2 ,84.7
[14]	Devanagari	Directional + curvature	SVM + MQDF	95.13

Methodology:

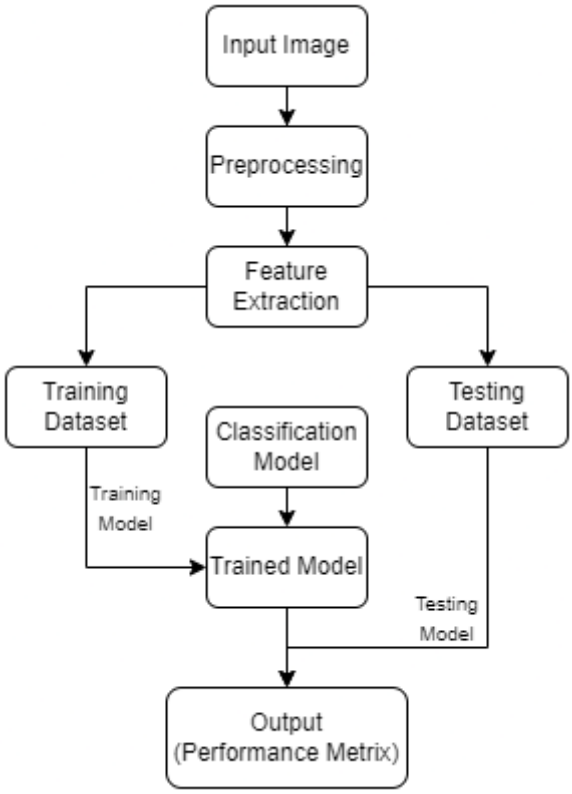

Fig 1 : Training of a Model

The research methodology adopted in this study employs a structured framework to assess and contrast the efficacy of various machine learning and deep learning models in the context of Optical Character Recognition (OCR) tasks. This framework is organized into several critical phases, which include data preparation, model selection, training, evaluation, and comparative analysis.

1. Data Preparation

- Two distinct datasets characterized by different resolutions (64x and 28x pixels) are chosen to reflect diverse image quality levels. The images undergo a series of preprocessing steps, which encompass normalization and label encoding. To enhance the robustness of the models, data augmentation techniques such as rotation, scaling, and the introduction of noise may be utilized. Subsequently, the datasets are partitioned into training, validation, and test subsets to facilitate an unbiased assessment.

2. Model Selection

- A variety of Convolutional Neural Network (CNN) architectures are developed, each featuring unique configurations that include differences in the number of convolutional layers, activation functions, and optimization algorithms. Traditional machine learning models, such as Support Vector Machines (SVM) and Random Forests, are also employed for comparative purposes. Furthermore, sequence-based models, including Long Short-Term Memory (LSTM) networks, are investigated for their capabilities in processing sequential data. Additionally, hybrid models that integrate encoder-

decoder frameworks are evaluated to capitalize on their advantages in feature extraction and sequence prediction.

3. Training

- The process of hyperparameter tuning is conducted to enhance model performance, taking into account variables such as dropout rates, the number of dense units, and the selection of optimizers. Each model is trained using the prepared datasets, with validation accuracy being closely monitored to mitigate the risk of overfitting. An early stopping mechanism is implemented to halt training if there is no improvement in validation accuracy over a predetermined number of epochs, thereby ensuring a more efficient training process.

4. Evaluation

- The primary metric employed to evaluate model performance is validation accuracy. Additionally, training and validation loss are tracked to gain insights into the learning process and to identify potential challenges such as underfitting or overfitting. In certain instances, confusion matrices are utilized to conduct a comprehensive analysis of classification performance, thereby revealing patterns of misclassification.

5. Comparison

- The performance of various models is assessed by examining validation accuracy, training duration, and complexity. Furthermore, the influence of dataset resolution on model efficacy is investigated by contrasting outcomes from the 64x and 28x datasets. This analysis aids in discerning which models excel under varying conditions and in establishing the most effective strategies for OCR tasks.

Experiment and Results:

Table 2: Experiment result of different models for 64x data and 28x data

Model Type	Best Model (64x)	Accuracy (64x)	Best Model (28x)	Accuracy (28x)	Remarks
CNN	Model 1	0.9882	Model 10	0.9763	CNNs perform well on both datasets, but higher resolution (64x) yields better results.
Encoder-Decoder	Encoder-Decoder	0.9781	Encoder-Decoder	0.981	Encoder-Decoder achieves high accuracy on both datasets, with slightly better performance on 28x.
SVM	SVM	0.9777	SVM	0.7252	SVM performs significantly better on the 64x dataset compared to 28x.
Random Forest	Random Forest	0.9538	Random Forest	0.9538	Random Forest shows consistent performance across both datasets.
LSTM	LSTM	0.0346	-	-	LSTM performs poorly on the 64x dataset and is not suitable for this task.

Table 2 compares the performance of various algorithms, including CNN, Encoder-Decoder, SVM, Random Forest, and LSTM, on two datasets with image resolutions of 64x and 28x pixels. The results of the experiment demonstrate the performance of different models on these datasets. CNNs achieve the highest accuracy, with Model 1 performing best at 98.82% on 64x and Model 10 at 97.63% on 28x. Encoder-Decoder models also perform well, showing a slight improvement on 28x. SVM shows a significant drop in accuracy from 97.77% on 64x to 72.52% on 28x, indicating its sensitivity to resolution. Random Forest maintains stable accuracy across both datasets at 95.38%. LSTM performs poorly, achieving only 3.46% accuracy on 64x, making it unsuitable for this task.

Table 3 : Selected CNN models based on different parameters

Model	Filters	Conv Layers	Activation	Dense Units	Dropout Rate	Optimizer
Model 1	128	4	ReLU	128	0.3	RMSprop
Model 2	32	3	ReLU	512	0.3	Adam
Model 3	128	2	ReLU	512	0.2	Adam
Model 4	64	4	Sigmoid	512	0.3	Adam
Model 5	128	3	Tanh	512	0.5	SGD
Model 6	64	3	Sigmoid	256	0.5	Adam
Model 7	64	2	Sigmoid	512	0.3	RMSprop
Model 8	64	3	Sigmoid	256	0.3	SGD
Model 9	64	4	Sigmoid	512	0.5	RMSprop
Model 10	64	2	Sigmoid	256	0.2	SGD

Table 3 shows different CNN models selected based on Filters , convolutional Layers , Activation Function , Dense Units , Dropout rates and Optimizers. This study investigates the performance of 10 commonly used CNN models with varying architectures, including different filter sizes, convolutional

layers, activation functions, dense units, dropout rates, and optimizers. These models were trained and evaluated on two datasets with resolutions of 64x and 28x to compare their effectiveness.

Fig 2 and Fig 3 compares the results of accuracy and loss of best performing models respectively. Both the figures are results after training the model with 28x bit images dataset and 64x images database respectively. The results indicate that Model 3 achieved the highest accuracy of 99.07% on both datasets, demonstrating strong generalization across different resolutions. Model 2 also performed well, achieving over 97% accuracy on both datasets. In contrast, Models 8, 9, and 10 exhibited significantly lower performance, with accuracies around 3.5%, suggesting that their configurations were ineffective for the OCR task. The findings highlight that models with optimized convolutional layers, suitable activation functions, and well-tuned dropout rates tend to yield superior results.

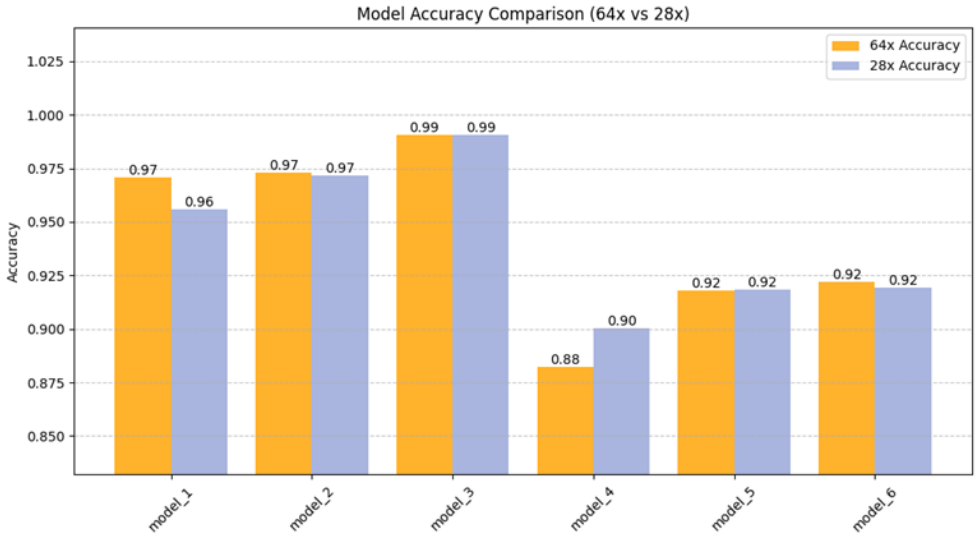

Fig 2 : Comparison of accuracy of best performing models

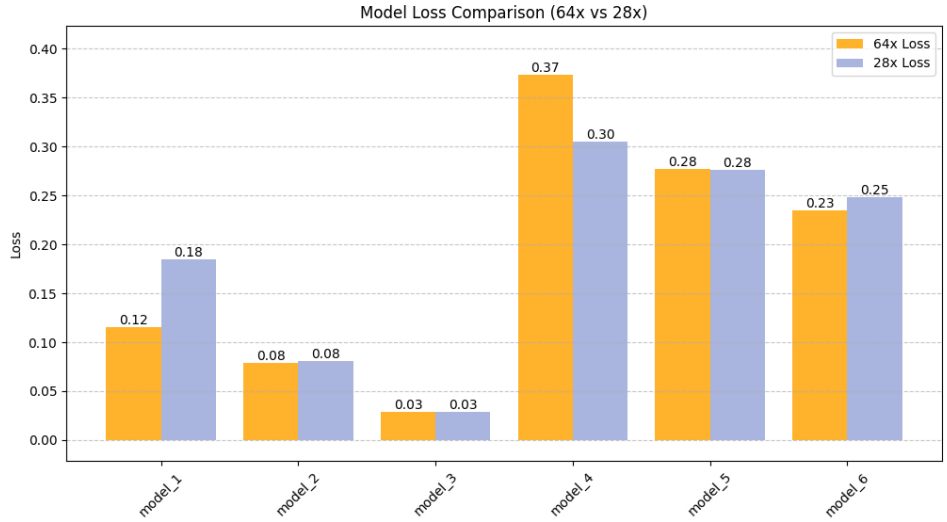

Fig 3: Comparison of loss of best performing models

The training history of the models across both the 64x and 28x datasets provides several significant insights. Models developed using the 64x dataset typically demonstrate superior accuracy and reduced loss in comparison to those utilizing the 28x dataset. For instance, Model 3 achieves a final accuracy of 0.9907 with a loss of 0.0287 when trained on the 64x dataset, underscoring the benefits of higher-resolution images for effective feature extraction. The trends in validation accuracy and loss mirror these findings, with models such as Model 3 and Model 2 exhibiting robust generalization capabilities on the 64x dataset.

Conversely, models trained on the 28x dataset show a slight decline in accuracy and an increase in loss. For example, Model 3 records a final accuracy of 0.9907 and a loss of 0.0291 on the 28x dataset, which, although commendable, is marginally inferior to its performance on the 64x dataset. The validation metrics for the 28x dataset also reflect a somewhat diminished generalization ability, indicating that lower-resolution images may hinder the model's capacity to learn complex patterns effectively.

In the comparative analysis of different models, Model 3 consistently excels across both datasets, achieving the highest accuracy and the lowest loss. This performance suggests that its architectural design, which includes considerations such as the number of filters, layers, and activation functions, is particularly well-suited for the OCR task. In contrast, models such as Model 8, Model 9, and Model 10 exhibit poor performance across both datasets, with final accuracies falling below 0.05 and high loss values, indicating potential inadequacies in their architectures or hyperparameter settings.

A notable trend in the training history shown in figure 4 and figure 5 indicates that models employing ReLU activation functions and optimizers like Adam or RMSprop tend to outperform those that utilize sigmoid activation and SGD optimizers. While higher-resolution datasets, such as the 64x, generally yield better results, well-optimized models, exemplified by Model 3, can still achieve commendable accuracy on lower-resolution datasets like the 28x.

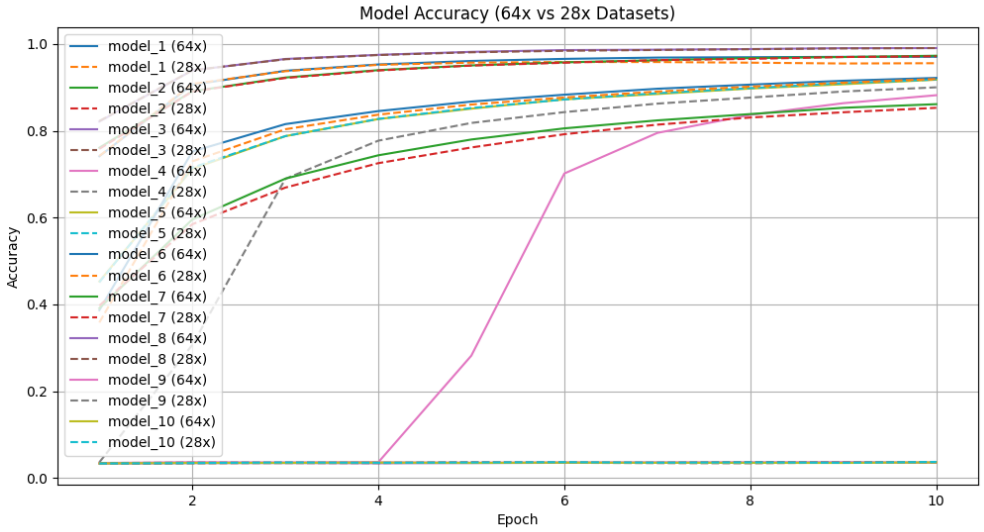

Fig 4: Model accuracy while training of all models with 28x dataset and 64x dataset

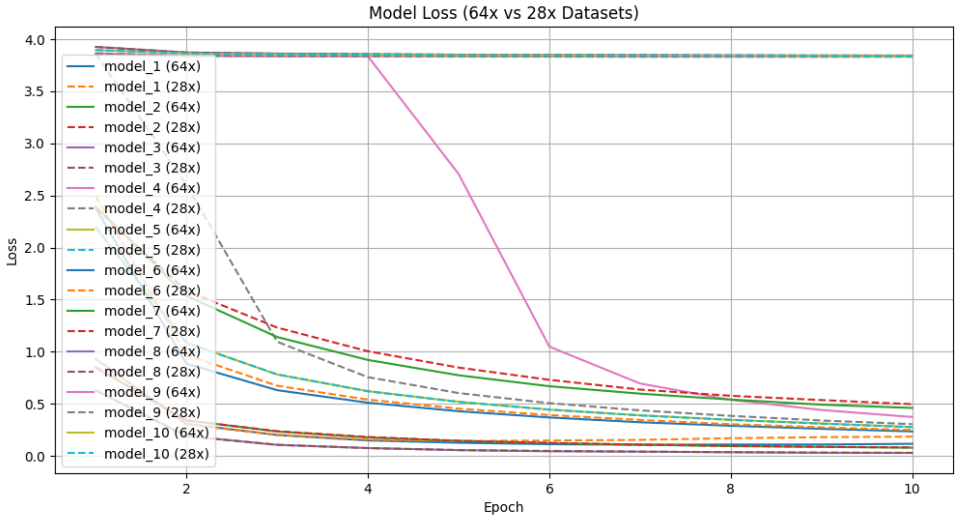

Fig 5: Model loss while training of all models with 28x dataset and 64x dataset

Conclusion

The training outcomes of models utilizing both the 64x and 28x datasets underscore the significant influence of image resolution on the performance of Optical Character Recognition (OCR) systems. Generally, models trained on the 64x dataset demonstrate superior accuracy and reduced loss, indicating that higher-resolution images facilitate more effective feature extraction. Notably, Model 3 consistently surpasses its counterparts, achieving a final accuracy of 0.9907 across both datasets, which suggests that a well-optimized architecture can maintain high performance even with lower-resolution inputs. Conversely, models such as Model 8, Model 9, and Model 10 show subpar performance, implying that certain architectural designs or hyperparameter settings may not be well-suited for this application. The analysis also indicates that the use of ReLU activation functions and optimizers like Adam or RMSprop significantly enhances performance when compared to sigmoid activation and SGD optimizers. While the 64x dataset yields superior results, a well-structured model can still achieve commendable accuracy with lower-resolution images, making it a practical option in scenarios with limited resources.

Future Scope of Research

Future investigations could aim to improve OCR performance by delving into deeper neural network architectures, incorporating attention mechanisms, and utilizing transfer learning techniques. Assessing the effects of data augmentation methods and synthetic data generation may enhance generalization, especially for datasets with lower resolutions. Furthermore, the development of lightweight models tailored for real-time applications on edge devices could broaden their practical applicability. Future research may also explore hybrid models that integrate Convolutional Neural Networks (CNNs) with transformers to capitalize on the advantages of both frameworks. Another promising avenue is the formulation of adaptive learning rate strategies to optimize training efficiency further. As advancements in deep learning continue, OCR models can be refined to achieve greater accuracy, improved generalization, and broader applicability across diverse fields.

References

[1] S. Mali, V. Karad, G. G. Rajput, and S. M. Mali, "Fourier Descriptor based Isolated Marathi Handwritten Numeral Recognition," Article in International Journal of Computer Applications, vol. 3, no. 4, pp. 975–8887, 2010, doi: 10.5120/724-1017.

[2] M. Raus and L. Kreft, "Reading car license plates by the use of artificial neural networks," Midwest Symposium on Circuits and Systems, vol. 1, pp. 538–541, 1995, doi: 10.1109/MWSCAS.1995.504495.

[3] U. Pal, T. Wakabayashi, N. Sharma, and F. Kimura, "Handwritten numeral recognition of six popular Indian scripts," Proceedings of the International Conference on Document Analysis and Recognition, ICDAR, vol. 2, pp. 749–753, 2007, doi: 10.1109/ICDAR.2007.4377015.

[4] P. Darshni, B. S. Dhaliwal, R. Kumar, V. A. Balogun, S. Singh, and C. I. Pruncu, "Artificial neural network based character recognition using SciLab," Multimed Tools Appl, vol. 82, no. 2, pp. 2517–2538, Jan. 2023, doi: 10.1007/S11042-022-13082-W/FIGURES/11.

[5] M. S. Sonawane and C. A. Dhawale, "Evaluation of character recognisers: Artificial neural network and nearest neighbour approach," Proceedings - 2015 IEEE International Conference on Computational Intelligence and Communication Technology, CICT 2015, pp. 129–132, Apr. 2015, doi: 10.1109/CICT.2015.30.

[6] A. G. Ramakrishnan, "Simultaneous Recognition of Tamil and Roman Scripts".

[7] D. Nasien, H. Haron, and S. S. Yuhaniz, "Support Vector Machine (SVM) for english handwritten character recognition," 2010 2nd International Conference on Computer Engineering and Applications, ICCEA 2010, vol. 1, pp. 249–252, 2010, doi: 10.1109/ICCEA.2010.56.

[8] Y. Babiker Hamdan, "Construction of Statistical SVM based Recognition Model for Handwritten Character Recognition," Journal of Information Technology and Digital World, vol. 03, no. 02, 2021, doi: 10.36548/jitdw.2021.2.003.

[9] N. M. Noor, M. Razaz, and P. Manley-Cooke, "Global geometry extraction for fuzzy logic based handwritten character recognition," Proceedings - International Conference on Pattern Recognition, vol. 2, pp. 513–516, 2004, doi: 10.1109/ICPR.2004.1334284.

[10] C. Koellner, M. Kurz, and E. Sonnleitner, "What Did You Mean? An Evaluation of Online Character Recognition Approaches," International Conference on Wireless and Mobile Computing, Networking and Communications, vol. 2019-October, Oct. 2019, doi: 10.1109/WIMOB.2019.8923384.

[11] A. Pal, & D. S.-I. J. of C. S., and undefined 2010, "Handwritten English character recognition using neural network," csjournals.com, Accessed: Mar. 20, 2023

[12] N. Arica and F. T. Yarman-Vural, "Optical character recognition for cursive handwriting," IEEE Trans Pattern Anal Mach Intell, vol. 24, no. 6, pp. 801–813, Jun. 2002, doi: 10.1109/TPAMI.2002.1008386.

[13] J. Zhang, G. Peng, H. Yang, C. Tan, Y. Tan, and H. Bai, "Real-Time Finger-Writing Character Recognition via ToF Sensors on Edge Deep Learning," Electronics 2023, Vol. 12, Page 685, vol. 12, no. 3, p. 685, Jan. 2023, doi: 10.3390/ELECTRONICS12030685.

[14] B. Saez-Mingorance, J. Mendez-Gomez, G. Mauro, E. Castillo-Morales, M. Pegalajar-Cuellar, and D. P. Morales-Santos, "Air-Writing Character Recognition with Ultrasonic Transceivers," Sensors 2021, Vol. 21, Page 6700, vol. 21, no. 20, p. 6700, Oct. 2021, doi: 10.3390/S21206700.

[15] T. T. Alemayoh, M. Shintani, J. H. Lee, and S. Okamoto, "Deep-Learning-Based Character Recognition from Handwriting Motion Data Captured Using IMU and Force Sensors," Sensors 2022, Vol. 22, Page 7840, vol. 22, no. 20, p. 7840, Oct. 2022, doi: 10.3390/S22207840.

[16] H. Zhang et al., "A Wearable Real-Time Character Recognition System Based on Edge Computing-Enabled Deep Learning for Air-Writing," J Sens, vol. 2022, 2022, doi: 10.1155/2022/8507706.

[17] N. A. Jebril, H. R. Al-Zoubi, and Q. Abu Al-Haija, "Recognition of Handwritten Arabic Characters using Histograms of Oriented Gradient (HOG)," Pattern Recognition and Image Analysis, vol. 28, no. 2, pp. 321–345, Apr. 2018, doi: 10.1134/S1054661818020141/METRICS.

[18] N. Lamghari, M. E. H. Charaf, and S. Raghay, "Hybrid Feature Vector for the Recognition of Arabic Handwritten Characters Using Feed-Forward Neural Network," Arab J Sci Eng, vol. 43, no. 12, pp. 7031–7039, Dec. 2018, doi: 10.1007/S13369-017-2969-1/METRICS.

[19] J. Cowell and F. Hussain, "Amharic character recognition using a fast signature based algorithm," Proceedings of the International Conference on Information Visualisation, vol. 2003-January, pp. 384–389, 2003, doi: 10.1109/IV.2003.1218014.

[20] R. Bajaj, L. Dey, and S. Chaudhury, "Devnagari numeral recognition by combining decision of multiple connectionist classifiers," vol. 27, pp. 59–72, 2002.

[21] U. Pal, S. Chanda, T. Wakabayashi, and F. Kimura, "Accuracy improvement of Devanagari character recognition combining SVM and MQDF Accuracy Improvement of Devnagari Character Recognition Combining SVM and MQDF," 2008, Accessed: Mar. 18, 2023.

[22] A. Bwatiramba, S. Venkataraman, and A. Ramjon, "Handwritten Character Recognition Using Deep Learning (Convolutional Neural Network)," vol. 14, no. 1, p. 2023, 2023, doi: 10.7176/CEIS/14-1-05.

[23] N. Singh, "An Efficient Approach for Handwritten Devanagari Character Recognition based on Artificial Neural Network," 2018 5th International Conference on Signal Processing and Integrated Networks, SPIN 2018, pp. 894–897, Sep. 2018, doi: 10.1109/SPIN.2018.8474282.

[24] A. Bhattacharyya, R. Chakraborty, S. Saha, S. Sen, R. Sarkar, and K. Roy, "A Two-Stage Deep Feature Selection Method for Online Handwritten Bangla and Devanagari Basic Character Recognition," SN Comput Sci, vol. 3, no. 4, pp. 1–16, Jul. 2022, doi: 10.1007/S42979-022-01157-2/METRICS.

[25] S. D. Connell, R. M. K. Sinha, and A. K. Jain, "Recognition of unconstrained on-line Devanagari characters," Proceedings - International Conference on Pattern Recognition, vol. 15, no. 2, pp. 368–371, 2000, doi: 10.1109/ICPR.2000.906089.

[26] D. Yadav, S. Sánchez-Cuadrado, and J. Morato, "Optical Character Recognition for Hindi Language Using a Neural-network Approach," J Inf Process Syst, vol. 9, no. 1, 2013, doi: 10.3745/JIPS.2013.9.1.117.

[27] A. Gaur and S. Yadav, "Handwritten Hindi character recognition using k-means clustering and SVM," 2015, ETTLIS 2015 - Proceedings, pp. 65–70, Feb. 2015, doi: 10.1109/ETTLIS.2015.7048173.

[28] N. Sharma, U. Pal, F. Kimura, and S. Pal, "Recognition of Off-Line Handwritten Devnagari Characters Using Quadratic Classifier," pp. 805–816, 2006, doi: 10.1007/11949619_72.

[29] M. Hanmandlu, O. V. Ramana Murthy, and V. K. Madasu, "Fuzzy model based recognition of handwritten Hindi characters," DICTA 2007, pp. 454–461, 2007, doi: 10.1109/DICTA.2007.4426832.

[30] A. Bhatti et al., "Recognition and Classification of Handwritten Urdu Numerals Using Deep Learning Techniques," Applied Sciences 2023, Vol. 13, Page 1624, vol. 13, no. 3, p. 1624, Jan. 2023, doi: 10.3390/APP13031624.

[31] X. Liu, B. Hu, Q. Chen, X. Wu, and J. You, "Stroke Sequence-Dependent Deep Convolutional Neural Network for Online Handwritten Chinese Character Recognition," IEEE Trans Neural Netw Learn Syst, vol. 31, no. 11, pp. 4637–4648, Nov. 2020, doi: 10.1109/TNNLS.2019.2956965.

[32] T. Fukumoto, T. Wakabayashi, F. Kimura, and Y. Miyake, "ACCURACY IMPROVEMENT OF HANDWRITTEN CHARACTER RECOGNITION BY GLVQ," 2004, Accessed: Mar. 20, 2023.

[33] S. Z. Gurbuz et al., "American Sign Language Recognition Using RF Sensing," IEEE Sens J, vol. 21, no. 3, pp. 3763–3775, Sep. 2020, doi: 10.1109/JSEN.2020.3022376.

[34] R. Babitha Lincy, J. Jency Rubia, C. Sherin Shibi, and M. Kavitha, "An Enhanced Deep Learning Model for Handwritten Tamil Character Identification," pp. 1518–1523, Mar. 2023, doi: 10.1109/ICSSIT55814.2023.10060920.

[35] Y. Yogesh, G. S. P. Ghantasala, and A. Priya, "Artificial Intelligence Based Handwriting Digit Recognition (HDR) - A Technical Review," Proceedings - IEEE International Conference on Device Intelligence, Computing and Communication Technologies, DICCT 2023, pp. 275–278, 2023, doi: 10.1109/DICCT56244.2023.10110186.