Telugu letters dataset and parallel deep convolutional neural network with a SGD optimizer model for TCR

Josyula Siva Phaniram, Mukkamalla Babu Reddy
Department of Computer Science Engineering, Krishna University, Machilipatnam, India

Article Info

Article history:

Received Dec 25, 2022
Revised May 11, 2023
Accepted Jun 30, 2023

Keywords:

Convolutional neural networks
Data pre-processing
Deep neural networks
Network optimization
Optical character recognition
Telugu character recognition
Telugu letter dataset

ABSTRACT

Because of the rapid growth in technology breakthroughs, including multimedia and cell phones, Telugu character recognition (TCR) has recently become a popular study area. It is still necessary to construct automated and intelligent online TCR models, even if many studies have focused on offline TCR models. The Telugu character dataset construction and validation using an Inception and ResNet-based model are presented. The collection of 645 letters in the dataset includes 18 Achus, 38 Hallus, 35 Othulu, 34×16 Guninthamulu, and 10 Ankelu. The proposed technique aims to efficiently recognize and identify distinctive Telugu characters online. This model's main pre-processing steps to achieve its goals include normalization, smoothing, and interpolation. Improved recognition performance can be attained by using stochastic gradient descent (SGD) to optimize the model's hyperparameters.

Corresponding Author:

Josyula Siva Phaniram
Department of Computer Science Engineering, Krishna University
Machilipatnam, Andhra Pradesh, India
Email: Phaniram.research@gmail.com

1. INTRODUCTION

One of the key modules in most optical character recognition (OCR) systems is character recognition, one of the pattern recognition study fields. The method typically starts with feature extraction and ends with classification [1]. Character recognition feature extraction involves converting a segmented character image into a real-valued feature vector that more accurately describes the character on the image. The capacity of features to discover characteristics that distinguish among participating classes aids classifiers in creating models with clearly defined decision limits [2]. The feature extraction algorithm significantly influences the character recognition process' accuracy. The segmented character images occasionally show the character in a slightly translated, rotated, or deformed state [3]. Images of documents in a high-quality state can use accurate character recognition technologies. Regarding real-time applications, document quality is essential and greatly impacts how well the recognition system works [4]. The key elements determining the document image quality are the characteristics of the document used, the contents within the document, and document deterioration. These characteristics subsequently impact recognition. Characters may appear broken or touching depending on the type of paper used, especially when printed documents are involved [5].

Non-standard fonts result in uneven spaces between the characters in printed documents and the issue of touching glyphs. In the case of handwritten documents, the content is created by a human; the uniformity of the space usage and the smoothness of the strokes influence the document's quality [6]. Another problem with handwritten papers is that some people simplify complex glyphs in the language script while writing them, which reduces inter-class diversity and increases intra-class variability [7]. Document aging and document digitization for processing may create many types of distortion in addition to problems in the paper used and defects that occurred during printing or writing. Some characters in documents may not be correctly recognized even by humans without contextual information due to the distortions and degradations that occurred to the document images [8]–[10]. Adopting a verification technique might be beneficial for such accuracy-sensitive applications where the recognition errors may be costly in some cases. The verification procedure assesses the classifier's or recognizer's performance and generates a trustworthy acceptance or rejection of the input pattern [11], [12].

Even in distortions, the recognition system should correctly identify the character. To develop a powerful character recognizer, the character image must be efficiently represented as an invariant picture. In most pattern recognition applications, deep learning (DL) approaches are cutting-edge. Convolutional neural network (CNN) architectures that are based on DL can learn invariant feature descriptors through the use of subsampling and trainable filter banks [13]–[15].

2. RELATED WORK

Historical texts usually contain a large number of dispersed characters, making it difficult to localise and distinguish between them using formal proposal and regression-based techniques. Yang *et al.* [16] published a unique approach known as a recognition guided detector that successfully detects precise Chinese characters in old texts. A detection network that uses this data to precisely localize each character and a recognition that provides context material about the text make up the two concurrently trained CNNs that make up the proposed reduced gradient descending (RGD). Two more datasets with character-level annotations were constructed to train and test the recommended method. The databases' contents are made up of scanned copies of the Tripitaka. Supported by text recognition with 97.25% accuracy.

Even though text arrangement created on natural language processing (NLP) has shown promising results and has a wide range of potential practical applications, including clinical medical value, the task of NLP for Chinese electronic medical records (CEMRs) has conventional less attention than English record data. The majority of the already accessible CEMRs are non-institutionalized texts with sloppy grammar, poor usage rates, and a propensity to mix patient symptoms, prescriptions, diagnoses, and other critical information. Zhang *et al.* [17] capsule network model for electronic medical record categorization uses a unique routing architecture and combines long short term memory (LSTM) and gated recurrent unit (GRU) models to extract intricate medical text elements. The model outperforms other baseline models by at least 4.1% and excels on the CEMRs dataset with an F1 score of 73.51%.

Given the ubiquity of handwritten documents in interpersonal interactions, character recognition (CR) of documents has enormous practical utility. Many different types of images can be transformed into editable, searchable, and analysable data thanks to the field of OCR. For the past ten years, researchers have been digitizing printed and handwritten texts using artificial intelligence (AI) and machine learning (ML) techniques. Memon *et al.* [18] examined character recognition to make investigation recommendations. Adhered to a predetermined review method and employed commonly used electronic databases. Observing in depth the selecting process for the study. There are 176 items in this systematic literature review (SLR) that were picked. In India, the bulk of the population speaks Hindi, however the language of most signboards is English. On a trip for business or pleasure, the travellers become bewildered by the numerous English-written signboards. They can rely on cell phones, which have grown in popularity in recent years, for the same functions.

According to Arafat *et al.* [19], they worked to develop a mobile application that can recognise the English text and symbols on a signboard image, detect and translate the content and symbols from English to Hindi, and then show the translated Hindi text back on the phone's screen. The system uses an English-to-Hindi lexicon for translation, a pre-trained faster regional CNNs for object detection, and tesseract OCR for text extraction. A means of communicating choices regarding the precise design, acquisition, procurement, building, and commissioning of a plant.

Kim *et al.* [20] provide a solution for the piping and instrumentation diagram (P&ID) picture text translation problem using DL technology. Pre-processing P&ID photos and storage of the recognition results are all steps in our suggested methodology. Think about how to identify symbols in high-density images that are different in size and complexity. When the model was tested on this dataset after it had been trained, the results were surprisingly good, with precision, and recall for symbols being 0.9718 and 0.9827, and for text being 0.9386 and 0.9175, respectively. Due to the rapidly expanding problem of financial tickets are putting financial accountants under increasing strain and wasting an unnecessary amount of personnel.

Zhang *et al.* [21] suggest an architecture of the financial ticket intelligent recognition system (FTIRS) that iteratively self-learns to address this problem. A functional financial accounting system must allow iterative updating and the flexibility of the algorithm model, both of which are supported by this framework. To increase its effectiveness and efficiency even more, developed an intelligent financial ticket data warehouse. The system can presently distinguish between 482 different sorts of financial tickets and has an autonomous iterative optimization process. As a result, the types of tickets that the system can recognise and their accuracy will grow as application processing times do as well. The system's value in business has been established. It can greatly boost financial accounting efficiency while also lowering the cost of recruiting accounting staff. Arabic, Chinese, and Hindi are other languages with cursive writing besides Latin. One of these scripts is Urdu. Urdu text makes it challenging to spot specific ligatures in scene shots and to locate them from natural scene photos.

In accordance with the method laid out by Arafat and Iqbal [22], Urdu ligatures are identified, their orientation is predicted, and they are recognised in outdoor pictures. Squeezenet, Googlenet, Resnet18, and Resnet50 have been integrated with the customised faster regions based CNN (FasterRCNN) algorithm for identification. To identify ligatures, a two-stream deep neural network (DNN) was employed. The common learning environment (CLE) annotation text was used to produce five sets of datasets containing 4.2K and 51K artificial images with embedded Urdu text for our testing, which evaluated a variety of functions including ligature detection. Additionally, time series deep neural network (TSDNN) was tested using 1,094 real images that had more than 12K Urdu characters. Evaluated and compared the capacity of all four detectors to locate or recognise Urdu text with average precision (AP). With an AP of 0.98, FasterRCNN, which is based on Resnet50 features, was shown to be the best detector.

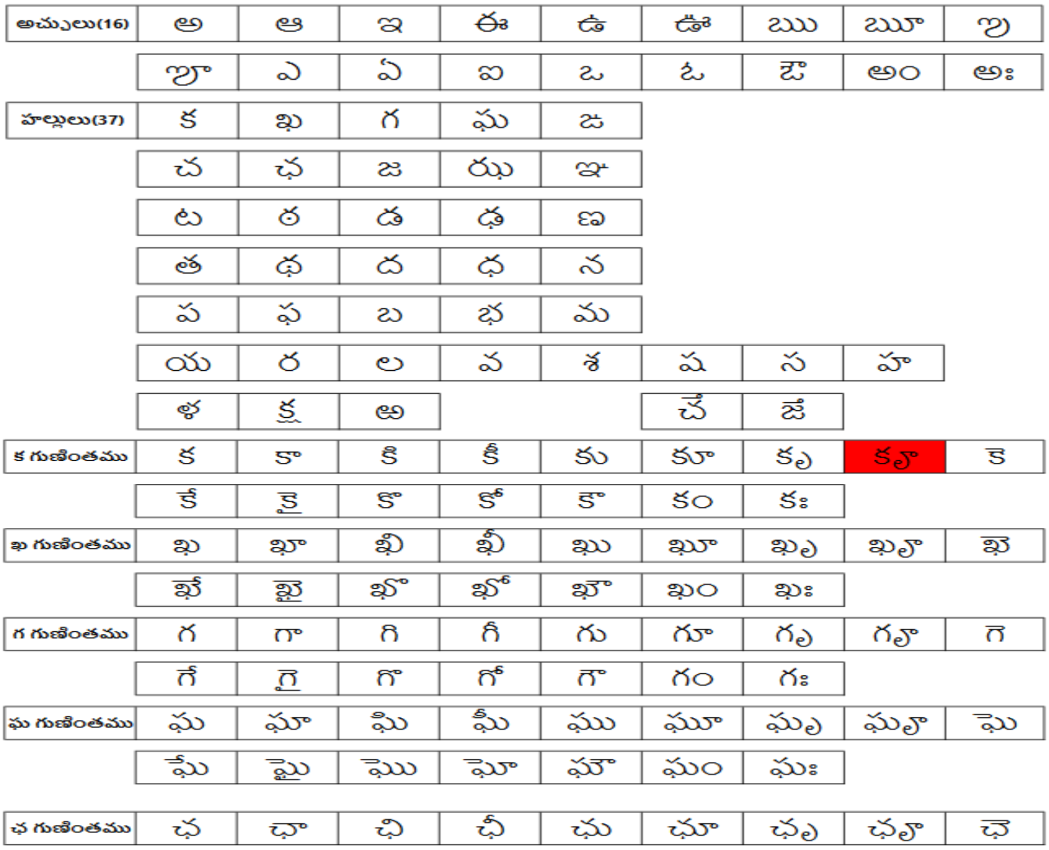
Oliveira *et al.* [23] addressed the non-overlapping camera vehicle identification problem in their study. Presenting the vehicle-rear dataset, a novel dataset for identifying vehicles is our main contribution. To investigate our dataset, our two-stream CNN makes use of the car's exterior and licence plate, two of the most recognisable and enduring aspects currently available. This initiative solves a serious problem: false alerts instigated by vehicles with identical designs or plates that are really similar [24]. A Siamese CNN can detect shape similarities in the first network stream by comparing two low-resolution car patches taken by two cameras. In the second stream, two high-resolution licence plate patches are used, and a CNN is used. To reach a decision, a set of entirely interconnected layers incorporate the features from both streams. OCR work as per state of art is as listed in Table 1.

Table 1. OCR as per state-of-art

Ref.	Pre-processing	Segmentation	Future extraction	Optimizer	Novelty/technique	Post processing
[24]	Residual identity block	Image segmentation	Gabor filter	Kernel self-optimization fisher classifier.	J&M model	---
[25]	Residual identity block	---	CNN feature extractor	Momentum optimizer	Hippocampus-heuristic character recognition network (HCRN), pseudo-Siamese network	---
[26]	Bilinear interpolation technique	---	ResNet and SEnet	Meta-optimization	customized recurrent neural network (CRNN)	Technique customs gate recurrent units
[27]	Niblack's method	---	Hamming network-hamming subnet, CNN subnet	Winner takes all algorithm (WTA)	complementary similarity measure (CSM) method, Convolutional neural networks	Similarity measure neural network (SMNN)
[28]	Data augmentation technology	Dropout and batch normalization	Separable convolution	Back propagation algorithm	ResNet's short connection structure	---
[29]	Binarization, skew correction, and noise removal	Projection segmentation	Novel adaptive sliding window	MLLR-based HMM adaptation techniques	Hidden Markov models	Viterbi decoding
[30]	Dilation and erosion	---	CNN (AlexNet)	Error-correcting output codes (ECOC)	CNN	Cross validation method
[31]	Scaling, noise reduction, centering, slanting, and skew estimation	Vertical projection profile analysis	Kernel method	Gradient descent optimization algorithm and Adam optimizer	DS-CapsNet and Caps-SoftPool method	Viterbi algorithm

3. TELUGU LETTER DATASET

In Telugu language there are total 645 letters i.e., 18 Achus, 38 Hallus, 35 Othulu, 34×16 Guninthamulu, and 10 Ankelu as shown in Figure 1. Here a dedicated dataset was designed for Telugu language with 645 classes. Within each category (Achus, Hallus, Othulu, and Guninthamulu), ensure that there's a diverse range of examples. This could include different writing styles, sizes, fonts, and orientations. Consider introducing variability between different categories to mimic real-world scenarios. For example, include combinations of Achus, Hallus, Othulu, and Guninthamulu in words to simulate how characters are used together.



అచ్చులు(16)	అ	ఆ	ఇ	ఈ	ఉ	ఊ	ఋ	ౠ	ఌ
	ౡ	ఎ	ఏ	ఐ	ఒ	ఓ	ఔ	అం	అః
హల్లులు(37)	క	ఖ	గ	ఘ	ఙ				
	చ	ఛ	జ	ఝ	ఞ				
	ట	ఠ	డ	ఢ	ణ				
	త	థ	ద	ధ	న				
	ప	ఫ	బ	భ	మ				
	య	ర	ల	వ	శ	ష	స	హ	
	ళ	క్ష	ఱ			ౘ	ౙ		
క గుణింతము	క	కా	కి	కీ	కు	కూ	కృ	కౄ	కె
	కే	కై	కొ	కో	కౌ	కం	కః		
ఖ గుణింతము	ఖ	ఖా	ఖి	ఖీ	ఖు	ఖూ	ఖృ	ఖౄ	ఖె
	ఖే	ఖై	ఖొ	ఖో	ఖౌ	ఖం	ఖః		
గ గుణింతము	గ	గా	గి	గీ	గు	గూ	గృ	గౄ	గె
	గే	గై	గొ	గో	గౌ	గం	గః		
ఘ గుణింతము	ఘ	ఘా	ఘి	ఘీ	ఘు	ఘూ	ఘృ	ఘౄ	ఘె
	ఘే	ఘై	ఘొ	ఘో	ఘౌ	ఘం	ఘః		
ఛ గుణింతము	ఛ	ఛా	ఛి	ఛీ	ఛు	ఛూ	ఛృ	ఛౄ	ఛె

Figure 1. Telugu letter considered for dataset

4. PARALLEL CONVOLUTIONAL NEURAL NETWORKS

The current standard in computer vision is deep CNNs (DCNNs). Despite all the work put into creating complex convolutional structures, it needs to be clarified how distinct the best CNNs are from one another. The current standard in computer vision (CV) is DCNNs. CNNs consistently place first in object recognition competitions and have been used for various visual tasks, including pose estimation, segmentation, object detection and localization, and visual saliency [25]. CNNs are a basis that may be used to instantiate numerous designs rather than a single design, which makes them less than ideal as a cognitive architecture. In contrast to their recognition rates, we compare CNNs based on how similar the final arrangement layers use the features to identify images.

4.1. ResNet

Contrarily, ResNet has a single-scale processing unit that is simpler, has numerous layers, and allows data to move over levels. A large portion of the CV works from the past five years may be viewed as a race between labs to develop the most compelling vision architecture using the deep network framework. Notwithstanding all the work put into creating convolutional structures, it needs to be clarified how distinct the best ones are from one another. ResNet was created at Microsoft in 2016 and then improved upon [26]. These numbers have been marginally enhanced by more modern architectures, including PNASNet. A large portion of current computer vision research can be seen as a race amongst research teams to develop the most effective deep network vision architecture.

ResNet which consists of two convolutional layers and a non-parameterized shortcut link that sends the output of the unaltered as shown in Figure 2, were introduced in 2015 by Ahmad *et al.* [27] deploying a 152-layer ResNet, significantly improved the challenge's state-of-the-art performance and proved that adding layers constantly improves recognition accuracy. It is possible to show that advances are still being made beyond 1,000 layers, which was previously impossible.

ResNet-v2 will be able to be improved further in 2016 [28]. This straightforward structural element has been included successfully in numerous additional DCNNs as listed in Table 2. These modules employ a concatenation of convolutional layers with various sizes and maximum pooling calculated from the same input.

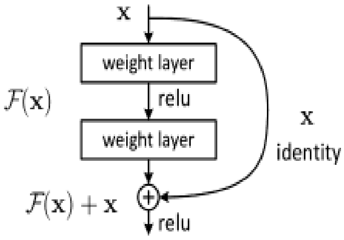

x
weight layer
relu
weight layer
$\mathcal{F}(\mathbf{x})$
x
identity
$\mathcal{F}(\mathbf{x}) + \mathbf{x}$
relu

Figure 2. ResNet shortcut connections

Table 2. Layers in ResNet

Conv. layer	O/P size	Layers				
		18	34	50	101	152
1	112×112	7×7,64, stride 2				
		3×3 max pool, stride 2				
2_X	56×56	$\left.\begin{matrix}3\times3,64\\3\times3,64\end{matrix}\right\}\times 2$	$\left.\begin{matrix}3\times3,64\\3\times3,64\end{matrix}\right\}\times 3$	$\left.\begin{matrix}1\times1,64\\3\times3,64\\1\times1,256\end{matrix}\right\}\times 3$	$\left.\begin{matrix}1\times1,64\\3\times3,64\\1\times1,256\end{matrix}\right\}\times 3$	$\left.\begin{matrix}1\times1,64\\3\times3,64\\1\times1,256\end{matrix}\right\}\times 3$
3_X	28×28	$\left.\begin{matrix}3\times3,128\\3\times3,128\end{matrix}\right\}\times 2$	$\left.\begin{matrix}3\times3,128\\3\times3,128\end{matrix}\right\}\times 4$	$\left.\begin{matrix}1\times1,128\\3\times3,128\\1\times1,512\end{matrix}\right\}\times 4$	$\left.\begin{matrix}1\times1,128\\3\times3,128\\1\times1,512\end{matrix}\right\}\times 4$	$\left.\begin{matrix}1\times1,128\\3\times3,128\\1\times1,512\end{matrix}\right\}\times 8$
4_X	14×14	$\left.\begin{matrix}3\times3,256\\3\times3,256\end{matrix}\right\}\times 2$	$\left.\begin{matrix}3\times3,256\\3\times3,256\end{matrix}\right\}\times 6$	$\left.\begin{matrix}1\times1,256\\3\times3,256\\1\times1,1024\end{matrix}\right\}\times 6$	$\left.\begin{matrix}1\times1,256\\3\times3,256\\1\times1,1024\end{matrix}\right\}\times 23$	$\left.\begin{matrix}1\times1,256\\3\times3,256\\1\times1,1024\end{matrix}\right\}\times 36$
5_X	7×7	$\left.\begin{matrix}3\times3,512\\3\times3,512\end{matrix}\right\}\times 2$	$\left.\begin{matrix}3\times3,512\\3\times3,512\end{matrix}\right\}\times 3$	$\left.\begin{matrix}1\times1,512\\3\times3,512\\1\times1,2048\end{matrix}\right\}\times 3$	$\left.\begin{matrix}1\times1,512\\3\times3,512\\1\times1,2048\end{matrix}\right\}\times 3$	$\left.\begin{matrix}1\times1,512\\3\times3,512\\1\times1,2048\end{matrix}\right\}\times 3$
	1×1	Average pool,1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

To develop inception, many of these components are combined. While the authors continued to optimise for classification performance, the modular architecture underwent several improvements. Using 1×1 convolutions, factorising n×n convolutional layers into stacked n×1 and 1×n layers, and batch normalization are important techniques, albeit no single architectural aspect is to blame for Inception's performance [29].

4.2. Inception

A CNN called inception separates processing by scale, blends the outcomes, and repeats. The inception family of CNNs includes inception. Inception's processing cost is also far cheaper than VGGNet or its more effective descendants. This has made it possible to use inceptions shown in Figure 3, in large data situations where huge quantities of data need to be processed affordably or where memory capability is naturally constrained, such as in mobile vision environments [30].

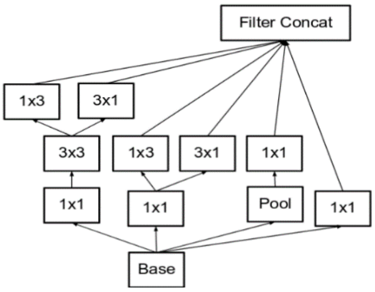


Figure 3. Inception module

By using specialized methods to target memory utilization or by using computational heuristics to optimize the execution of specific activities, it is possible to partially minimize these concerns. These techniques, however, increase complexity. The efficiency difference could also be widened by using similar techniques to improve the inception architecture.

5. OPTIMIZATION

Several gradient descent (GD) methods are obtainable in the works that can be used to systematically address the issue of "local" minima, as previously discussed. In this section is a description of the approaches that are most frequently utilized. Batch gradient descent (BGD) determines the error, but the model is updated only after all the training samples have been assessed. A training epoch is a name given to this entire process, which resembles a cycle [31]. The benefits of computational effectiveness and stable convergence, as well as the drawbacks of BGD, are each epoch must have convergence to local minima and access to the entire training dataset. In contrast to BGD, stochastic gradient descent (SGD) calculates the error. In other words, it individually adjusts the settings.

The main advantages are due to sequential data processing and generally faster than BGD, the drawbacks of SGD are the detailed rate of development. Frequent updates are costly computationally and introduce noise into gradients that hinder convergence. SGD has difficulty navigating ravines, which are frequently found near local optimums and are defined as regions where the surface slopes considerably more sharply in one dimension than another. A function's local minimum can be found using the optimization technique known as GD [32]. The weights are iteratively updated in backpropagation to reduce the error function. The GD optimization's calculation time can increase dramatically when training sets are big since each iteration requires computing every sample's outputs, errors, and gradients. In neural networks, SGD is thus virtually always favoured over GD using, the weights are updated.

$$W_{ij}^{\tau+1} = W_{ij}^{\tau} + \Delta\, W_{ij}^{\tau}, \tag{1}$$

Where W_{ij} weight update and τ GD iteration. The two most popular SGD variations are batch learning and online learning. For each input (k) in on-line learning, the weights are changed for every input $\alpha(k)$, *i.e.*, in (1) as (2):

$$\Delta\, W_{ij} = -\eta \frac{\partial E}{\partial W_{ij}}^{(k)} = -\eta \delta_j^{l} \alpha_j^{l-1(k)} \tag{2}$$

where η the rate of learning, the network's hyperparameter controls how quickly things change. In batch learning, the weight gradients are calculated, and the error for a batch of K inputs A.

$$\Delta\, W_{ij} = -\eta \frac{\partial E}{\partial W_{ij}} = -\frac{\eta}{K} \sum_{k=1}^{K} \delta_j^{l(k)} \alpha_j^{l-1(k)} \tag{3}$$

Being caught in a local minimum that is too far from the wanted global least is a typical problem with gradient-based optimization techniques. Online learning helps to avoid such local minima because the stochastic error surface is noisy. Batch learning reduces noise and is more likely to become stuck in local minima by averaging the gradients. However, batch training is frequently favored because it can be carried out quite well with modern machines. Additionally, training techniques have been created to avoid local minima, rendering online learning unnecessary. Hyperparameters took into account for SGD optimization [33].

```
------------------------------------------------------------------------------------------------------------
optimizers.SGD
{
learning_rate=0.01, momentum=0.0, beta_1=0.9, beta_2=0.999, epsilon=1e-07,
nesterov=False, name="SGD",
}
------------------------------------------------------------------------------------------------------------
```

The momentum approach in optimization algorithms, such as SGD, plays a crucial role in mitigating the issues of convergence to local minima during the training of neural networks. It accomplishes this by reducing the oscillations or swings in weight updates over successive iterations. By introducing a momentum term, which is essentially a moving average of past gradients, the algorithm gains inertia and tends to continue in the direction of the previously accumulated gradients. This means that when the gradient changes direction, as it often does in complex loss landscapes, the momentum helps the optimizer to carry on and escape shallow local minima. In essence, momentum smoothes out the optimization path, allowing the algorithm to navigate more efficiently through the optimization space, ultimately speeding up convergence and improving the chances of finding a better global minimum.

6. IMPLEMENTATION

It was demonstrated to recognize Telugu characters using a parallel DCNN with a SGD optimizer (PDCNN-SGD) as shown in Figure 4. Inception and ResNet employ fully connected (FC) classifiers to assign labels to images based on the retrieved features after extracting features from images using convolutional architecture. The architectures and the number of features retrieved by the two systems differ; ResNet produces 2,048 components per image, whereas inception produces 1,536 [34].

However, you will see that the features recovered by Inception and ResNet are extremely similar in that the affine mapping of one predicts the other. This implies that despite their structural differences, inception and ResNet utilize essentially the same aspects of images. While they might extract the same features, inception seems to accomplish so more robustly than ResNet. According to a further review of our results, it might extract a few additional properties. It is unexpected to learn that affine transformations connect ResNet and inception features.

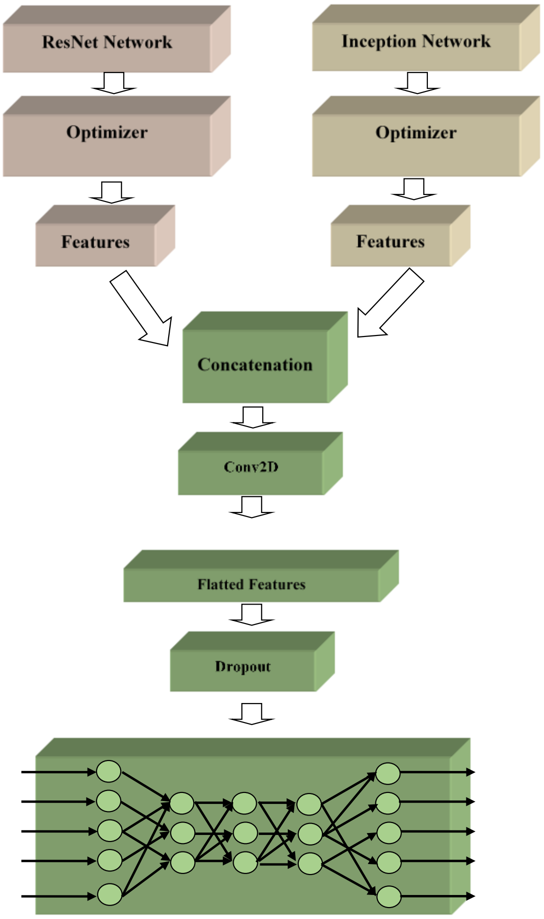


Figure 4. PDCNN-SGD optimizer for Telugu character recognition (TCR)

Because CNNs can learn intricate non-linear aspects of images, they have completely changed the area of computer vision. The structural variations across the systems lead one to believe that the non-linear functions used by the various systems assume different shapes. However, as their features are linear changes of one another, Inception and ResNet appear to extract similar qualities from images. Their training algorithms appear to do solutions-finding hill-climbing in entirely different environments.

The two systems identical performance is explained by this affine relationship, which, in our opinion, also has wider ramifications. It implies that the content of the training images, rather than the specifics of Inception and ResNet's neural architectures, drives the features that are retrieved by those systems. If this is the case, many complex CNNs should behave similarly. We used 50 and 100 epochs as variations on the number of epochs in the trials we did as shown in Figures 5 and 6.

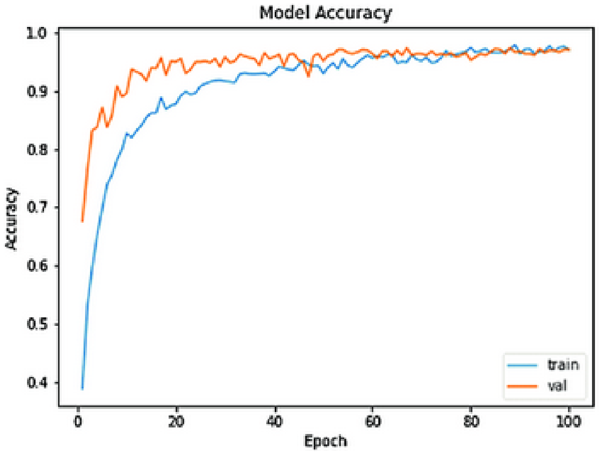


Figure 5. Accuracy graph analysis of PDCNN-SGD technique

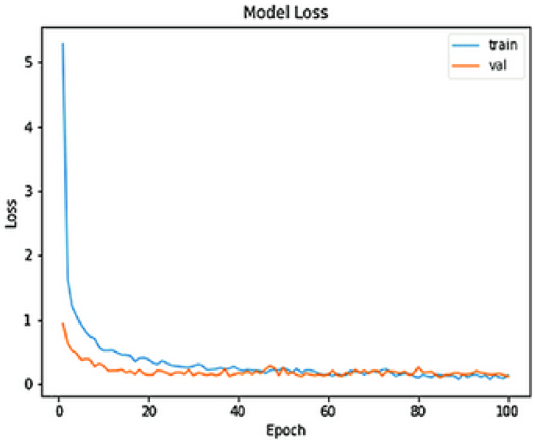


Figure 6. Loss graph analysis of PDCNN-SGD technique

Combining ResNet and inception, two prominent DL architectures, offers superior performance due to their complementary strengths. ResNet's ability to handle very deep networks and inception's efficient multi-scale feature extraction capabilities make them an ideal pair. This ensemble reduces overfitting, enhances generalization, and provides increased accuracy, making it a powerful choice for various computer vision tasks, including image classification, object detection, and semantic segmentation. According to experimental findings, our suggested method performs better and is appropriate for TCR as shown in Figure 7.

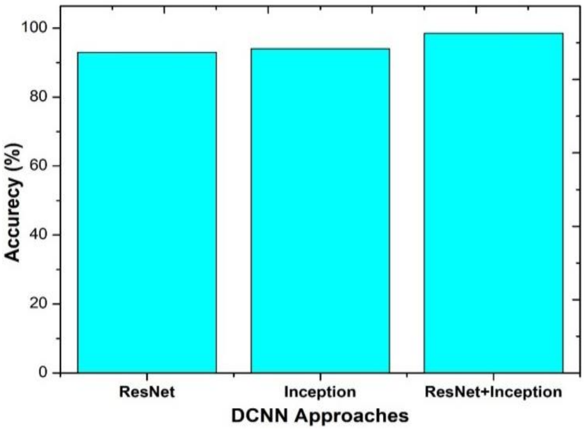


Figure 7. Accuracy of models with different DCNN approaches

Due to three characteristics, experimental findings showed that the suggested hybrid model. While the success of most traditional classifiers depends mainly on the retrieval of suitable hand-crafted features extractor, which is a laborious and time-consuming operation, the prominent features of the images can be automatically retrieved by the hybrid model. Given that ResNet and inception are the most widely used and effective classifiers, the hybrid model incorporates the strengths of both techniques. Comparing the hybrid model to the CNN classification model only slightly increases the complexity of the decision-making process.

7. CONCLUSION

The PDCNN-SGD model, designed for recognizing and classifying Telugu characters in handwritten text, is a complex DL architecture consisting of several stages of operations to achieve its objective effectively. The dataset comprises a total of 645 letters, including 18 Achus, 38 Hallus, 35 Othulu, 34×16 Guninthamulu, and 10 Ankelu. In the feature extraction stage, both ResNet and inception architectures are employed to extract rich and diverse feature representations from the input data. The SGD-based hyperparameter optimization process systematically tunes critical hyperparameters, such as learning rate, batch size, and weight decay, to find the optimal configuration for training the model, ensuring its readiness to achieve high accuracy in TCR tasks.

REFERENCES

[1] N. P. Sutramiani, N. Suciati, and D. Siahaan, "MAT-AGCA: multi augmentation technique on small dataset for balinese character recognition using convolutional neural network," *ICT Express*, vol. 7, no. 4, pp. 521–529, Dec. 2021, doi: 10.1016/j.icte.2021.04.005.

[2] G. Huang, X. Luo, S. Wang, T. Gu, and K. Su, "Hippocampus-heuristic character recognition network for zero-shot learning in Chinese character recognition," *Pattern Recognition*, vol. 130, p. 108818, Oct. 2022, doi: 10.1016/j.patcog.2022.108818.

[3] B. R. Kavitha and C. Srimathi, "Benchmarking on offline handwritten tamil character recognition using convolutional neural networks," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 4, pp. 1183–1190, Apr. 2022, doi: 10.1016/j.jksuci.2019.06.004.

[4] Y. Zheng, J. Zhao, Y. Chen, C. Tang, and S. Yu, "3D mesh model classification with a capsule network," *Algorithms*, vol. 14, no. 3, Mar. 2021, doi: 10.3390/a14030099.

[5] N. Saqib, K. F. Haque, V. P. Yanambaka, and A. Abdelgawad, "Convolutional-neural-network-based handwritten character recognition: an approach with massive multisource data," *Algorithms*, vol. 15, no. 4, Apr. 2022, doi: 10.3390/a15040129.

[6] G.-S. Lin, J.-C. Tu, and J.-Y. Lin, "Keyword detection based on RetinaNet and transfer learning for personal information protection in document images," *Applied Sciences*, vol. 11, no. 20, Jan. 2021, doi: 10.3390/app11209528.

[7] Y.-Q. Li, H.-S. Chang, and D.-T. Lin, "Large-scale printed chinese character recognition for ID cards using deep learning and few samples transfer learning," *Applied Sciences*, vol. 12, no. 2, Jan. 2022, doi: 10.3390/app12020907.

[8] M. Z. Rehman, N. M. Nawi, M. Arshad, and A. Khan, "Recognition of cursive pashto optical digits and characters with trio deep learning neural network models," *Electronics*, vol. 10, no. 20, Jan. 2021, doi: 10.3390/electronics10202508.

[9] H. Butt, M. R. Raza, M. J. Ramzan, M. J. Ali, and M. Haris, "Attention-based CNN-RNN Arabic text recognition from natural scene images," *Forecasting*, vol. 3, no. 3, Sep. 2021, doi: 10.3390/forecast3030033.

[10] M. Mudhsh and R. Almodfer, "Arabic handwritten alphanumeric character recognition using very deep neural network," *Information*, vol. 8, no. 3, Sep. 2017, doi: 10.3390/info8030105.

[11] C. Boufenar, A. Kerboua, and M. Batouche, "Investigation on deep learning for off-line handwritten Arabic character recognition," *Cognitive Systems Research*, vol. 50, pp. 180–195, Aug. 2018, doi: 10.1016/j.cogsys.2017.11.002.

[12] M. M. Khan, M. S. Uddin, M. Z. Parvez, and L. Nahar, "A squeeze and excitation ResNeXt-based deep learning model for Bangla handwritten compound character recognition," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 6, Part B, pp. 3356–3364, Jun. 2022, doi: 10.1016/j.jksuci.2021.01.021.

[13] S. Roy, N. Das, M. Kundu, and M. Nasipuri, "Handwritten isolated Bangla compound character recognition: a new benchmark using a novel deep learning approach," *Pattern Recognition Letters*, vol. 90, pp. 15–21, Apr. 2017, doi: 10.1016/j.patrec.2017.03.004.

[14] N. Shaffi and F. Hajamohideen, "uTHCD: a new benchmarking for Tamil handwritten OCR," *IEEE Access*, vol. 9, pp. 101469–101493, 2021, doi: 10.1109/ACCESS.2021.3096823.

[15] A. Rasheed, N. Ali, B. Zafar, A. Shabbir, M. Sajid, and M. T. Mahmood, "Handwritten urdu characters and digits recognition using transfer learning and augmentation with AlexNet," *IEEE Access*, vol. 10, pp. 102629–102645, 2022, doi: 10.1109/ACCESS.2022.3208959.

[16] H. Yang, L. Jin, W. Huang, Z. Yang, S. Lai, and J. Sun, "Dense and tight detection of chinese characters in historical documents: datasets and a recognition guided detector," *IEEE Access*, vol. 6, pp. 30174–30183, 2018, doi: 10.1109/ACCESS.2018.2840218.

[17] Q. Zhang, Q. Yuan, P. Lv, M. Zhang, and L. Lv, "Research on medical text classification based on improved capsule network," *Electronics*, vol. 11, no. 14, Jan. 2022, doi: 10.3390/electronics11142229.

[18] J. Memon, M. Sami, R. A. Khan, and M. Uddin, "Handwritten optical character recognition (OCR): a comprehensive systematic literature review (SLR)," *IEEE Access*, vol. 8, pp. 142642–142668, 2020, doi: 10.1109/ACCESS.2020.3012542.

[19] S. Y. Arafat, N. Ashraf, M. J. Iqbal, I. Ahmad, S. Khan, and J. J. P. C. Rodrigues, "Urdu signboard detection and recognition using deep learning," *Multimed Tools Appl*, vol. 81, no. 9, pp. 11965–11987, Apr. 2022, doi: 10.1007/s11042-020-10175-2.

[20] H. Kim *et al.*, "Deep-learning-based recognition of symbols and texts at an industrially applicable level from images of high-density piping and instrumentation diagrams," *Expert Systems with Applications*, vol. 183, p. 115337, Nov. 2021, doi: 10.1016/j.eswa.2021.115337.

[21] H. Zhang, Q. Zheng, B. Dong, and B. Feng, "A financial ticket image intelligent recognition system based on deep learning," *Knowledge-Based Systems*, vol. 222, p. 106955, Jun. 2021, doi: 10.1016/j.knosys.2021.106955.

[22] S. Y. Arafat and M. J. Iqbal, "Urdu-text detection and recognition in natural scene images using deep learning," *IEEE Access*, vol. 8, pp. 96787–96803, 2020, doi: 10.1109/ACCESS.2020.2994214.

[23] I. O. D. Oliveira, R. Laroca, D. Menotti, K. V. O. Fonseca, and R. Minetto, "Vehicle-rear: a new dataset to explore feature fusion for vehicle identification using convolutional neural networks," *IEEE Access*, vol. 9, pp. 101065–101077, 2021, doi: 10.1109/ACCESS.2021.3097964.

[24] A. F. D. S. Neto, B. L. D. Bezerra, E. B. Lima, and A. H. Toselli, "HDSR-flor: a robust end-to-end system to solve the handwritten digit string recognition problem in real complex scenarios," *IEEE Access*, vol. 8, pp. 208543–208553, 2020, doi: 10.1109/ACCESS.2020.3039003.

[25] S.-H. Lee, W.-F. Yu, and C.-S. Yang, "ILBPSDNet: based on improved local binary pattern shallow deep convolutional neural network for character recognition," *IET Image Processing*, vol. 16, no. 3, pp. 669–680, 2022, doi: 10.1049/ipr2.12226.

[26] S. Gorla, L. B. M. Neti, and A. Malapati, "Enhancing the performance of telugu named entity recognition using gazetteer features," *Information*, vol. 11, no. 2, Feb. 2020, doi: 10.3390/info11020082.

[27] I. Ahmad, S. A. Mahmoud, and G. A. Fink, “Open-vocabulary recognition of machine-printed Arabic text using hidden Markov models,” *Pattern Recognition*, vol. 51, pp. 97–111, Mar. 2016, doi: 10.1016/j.patcog.2015.09.011.

[28] M. B. Bora, D. Daimary, K. Amitab, and D. Kandar, “Handwritten character recognition from images using CNN-ECOC,” *Procedia Computer Science*, vol. 167, pp. 2403–2409, Jan. 2020, doi: 10.1016/j.procs.2020.03.293.

[29] A. K. O. Mohammed and S. Poruran, “OCR-Nets: variants of pre-trained CNN for Urdu handwritten character recognition via transfer learning,” *Procedia Computer Science*, vol. 171, pp. 2294–2301, Jan. 2020, doi: 10.1016/j.procs.2020.04.248.

[30] Hubert, P. Phoenix, R. Sudaryono, and D. Suhartono, “Classifying promotion images using optical character recognition and naïve bayes classifier,” *Procedia Computer Science*, vol. 179, pp. 498–506, Jan. 2021, doi: 10.1016/j.procs.2021.01.033.

[31] M. Jangid and S. Srivastava, “Handwritten devanagari character recognition using layer-wise training of deep convolutional neural networks and adaptive gradient methods,” *Journal of Imaging*, vol. 4, no. 2, Feb. 2018, doi: 10.3390/jimaging4020041.

[32] S. Patil *et al.*, “Enhancing optical character recognition on images with mixed text using semantic segmentation,” *Journal of Sensor and Actuator Networks*, vol. 11, no. 4, Dec. 2022, doi: 10.3390/jsan11040063.

[33] D. Mengu, Y. Luo, Y. Rivenson, and A. Ozcan, “Analysis of diffractive optical neural networks and their integration with electronic neural networks,” *IEEE Journal of Selected Topics in Quantum Electronics*, vol. 26, no. 1, pp. 1–14, Jan. 2020, doi: 10.1109/JSTQE.2019.2921376.

[34] I. A. D. Williamson, T. W. Hughes, M. Minkov, B. Bartlett, S. Pai, and S. Fan, “Reprogrammable electro-optic nonlinear activation functions for optical neural networks,” *IEEE Journal of Selected Topics in Quantum Electronics*, vol. 26, no. 1, pp. 1–12, Jan. 2020, doi: 10.1109/JSTQE.2019.2930455.

BIOGRAPHIES OF AUTHORS

Josyula Siva Phaniram has completed master of science in computer science from Nagarjuna University and master of technology in information technology from Karnataka State University. He is currently pursuing Ph.D. in computer science at Krishna University, Machilipatnam. His research interest includes artificial intelligence, machine learning, and deep learning. He can be contacted at email: phaniram.research@gmail.com.

Dr. Mukkamalla Babu Reddy has completed his post graduation and doctoral degree from Acharya Nagarjuna University, Andhra Pradesh. He is activity involved in teaching and research for more than two decades. He has published more than 60 research papers in reputed journals and guiding Ph.D. scholars in computer science and engineering domain. At present, he is working as principal, University College of Engineering and Technology, Krishna University, Machilipatnam, Andhra Pradesh, India. He can be contacted at email: m_babureddy@yahoo.com.