

## Research Article

# Enhancing the Power of CNN Using Data Augmentation Techniques for Odia Handwritten Character Recognition

Mamatarani Das ,<sup>1</sup> Mrutyunjaya Panda ,<sup>1</sup> and Shreela Dash ,<sup>2</sup>

<sup>1</sup>*Utkal University, Bhubaneswar, India*

<sup>2</sup>*Silicon Institute of Technology, Bhubaneswar, India*

Correspondence should be addressed to Mamatarani Das; [mamataparida2005@gmail.com](mailto:mamataparida2005@gmail.com)

Received 12 September 2022; Revised 5 December 2022; Accepted 7 December 2022; Published 22 December 2022

Academic Editor: Marco Roccati

Copyright © 2022 Mamatarani Das et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The performance of any machine learning model largely depends on the type of input data provided. The higher the volume and variety of the data, the better the machine learning models get trained, thereby producing more accurate results. However, it is a challenging task to get high volume of data in some cases containing enough variety. Handwritten character recognition for Odia language is one of them. NITROHCS v1.0 for handwritten Odia characters and the ISI image database for handwritten Odia numerals are the standard Odia language datasets available for the research community. This paper shows the performance of five different machine learning models that uses a convolutional neural network to identify handwritten characters in response to handwritten datasets that are manipulated and expanded using several augmentation techniques to create variation and increase the volume of the data in the given dataset. These models, with the augmentation techniques discussed in the paper, even lead to a further increase in accuracy by approximately 1% across the models. The claims are supported by the results from the experiments done on the proposed convolutional neural network models on standard available Odia character and numeral data set.

## 1. Introduction

By using their eyes and brains, humans can see and visually feel the world around them. Making computers capable of perceiving and processing images in the identical way that individuals can is the goal of the vision of a computer. The domain of computer vision has produced a number of techniques for image recognition. From a given sensory input, hierarchical layers of representation are learned by a deep neural network (DNN) to perform pattern recognition [1–3]. These deep architectures have recently shown extremely remarkable outcomes, often on par with the results of humans [4, 5]. However, despite more than five decades of intensive research, the computer's reading ability is still far below that of humans. Most optical character recognition (OCR) technologies are still unable to read deteriorated documents or handwritten notes.

In the past, handwriting recognition algorithms relied heavily on handcrafted features and needed extensive prior information. Based on these requirements, it is difficult to train an optical character recognition (OCR) system and comparatively produces lesser classification accuracy. Deep learning methods are now at the forefront of handwriting recognition research, which has produced some outstanding achievements in recent years. Nonetheless, the growing amount of handwritten dataset, combined with the availability of massive computational power, results in an increase in recognition accuracy, inspiring researchers to continue their research in the field of character recognition using convolutional neural networks (CNN).

CNNs are particularly effective at extracting the various features of handwritten characters and recognizing their structure automatically. However, there are some limitations, such as the fact that CNN models frequently require massive amounts of data for training. Data augmentation

techniques are used to generate different replicas of the same data, introducing variants as well as boosting the data for training, in addition to artificially enhancing the volume of an existing dataset. A deep learning model trained on augmented images along with the original images outperforms a deep learning model that is only trained on the original images. Other than this, in general, the augmentation reduces the cost of collecting data when it is scarce and enhances the generalization ability of the data models.

If we see the state-of-the-art in HCR for Odia language, there is very less amount of contributions towards this area of research compared to other Indian languages; due to the roundish shape of Odia character, the presence of large number of modified and compound characters and similarity between different characters makes this language very hard to create a satisfactory classifier. So in our proposed CNN model, we are trying to achieve human-like accuracy for Odia HCR.

The proposed work has two objectives:

O1: one goal is to attain comparable accuracy for handwritten Odia digit and character recognition using a regularized CNN architecture.

O2: another goal is to look into various augmentation methods and how they affect the proposed CNN architecture's performance.

So, this work's main contributions are as follows:

C1: it is a thorough evaluation of five different baseline models proposed by varying the number of features in the convolutional layers and the number of units in the dense layer from one architecture to the next.

C2: to avoid overfitting the models, L2 regularization and spatial dropout were added to the models to enhance accuracy and the performance of the baseline and regularized models analyzed.

C3: different augmentation techniques are applied to the databases used for our experimentation to create variation and increase the volume of the data. A set of the best data augmentation techniques is proposed and supported by the experimental results.

The rest of the paper is as follows: The related work is detailed in Section 2 and Section 3 presents the technique, which includes the datasets utilized for the research and the five distinct CNN architectures that are used in this research for handwritten character recognition. Techniques for image enhancement are covered in Section 4. The findings of the experiments are the subject of Section 5, and the conclusion is in Section 6.

## 2. Related Work on Handwritten Character Recognition

Odia (previously Oriya) is a popular language in India recognized by the constitution as well as the official language of the state of Odisha. Handwritten character recognition (HCR), online or offline, postal-address interpretation, writer recognition, signature verification, real-time

handwriting recognition, bank-check/cheque processing, or note preparation are only a few of the ongoing study fields where deep learning produces better accuracy. Several studies have been conducted in the domain of optical character recognition in several languages [6, 7], but progress in the Odia language has been limited. The authors of [8] analyze various approaches for handwritten character recognition using a standard handwritten digit recognition test, and convolutional neural networks (CNNs) have been found to outperform all other techniques when dealing with the variability of 2-D shapes. The authors of [9] classified the printed Odia characters from the ISI Kolkata Dataset and obtained an accuracy of 96.3%. The preprocessing techniques used by authors were skew detection and correction, followed by line, word, and character segmentation. Stroke and run-number-based features, along with the features obtained from the concept of a water reservoir, were used, and the decision tree classifier was used for the classification task. In [10], binarization, skeletonization by chain coding, noise removal, and segmentation were the preprocessing techniques used by authors for the Odia Digit Database, NIT Rourkela, and they obtained an accuracy of 96.08% using the Finite Automata classifier, whereas in [11] binary external symmetry axis constellation (BESAC) features were used on the IITBBS Odia character database having 7800 data samples. The accuracy of random forest classifier was 89.92%, SVM classifier was 93.77%, and K-Neighbor classifier was 95.01%. An ensemble way of selecting features as well as classification of Odia characters has been proposed by [12]. Husnain et al. [13–17] contributed their work on the identification of Odia alphabets and digits. Researchers used neural networks and other deep learning approaches to contribute to the field of character classification, as documented in [13, 18–20]. In [21], the authors contributed to work on image augmentation based on generative adversarial networks (GANs) on an ISI Kolkata handwritten dataset of Latin, Bangla, Devanagari, and Oriya languages. GAN is a method for developing artificial sample images for a database that does not necessitate prior knowledge of the probable differences between samples and has obtained an accuracy of 97.31% on the Oriya (Odia) character set. Similar to Odia HCR, if we investigate HCR for other regional Indian languages such as Bengali, Devanagari, or Telugu, most of the works involve a machine learning approach associated with handcrafted feature extraction followed by classification [22]. Here, the authors propose a feature extraction technique to extract features to classify the Bangla compound characters. The feature vector of 180 length is constructed from the longest run feature (LRF), the histogram of oriented gradients (HOG) feature, and the diagonal feature. The extracted features were used to train an SVM classifier, which achieved 88.73% accuracy. The authors of [23] proposed a method for digit recognition called "Celled Projection" that partitions the image and computes the projection of each section, and the k-NN classifier achieved an accuracy of 94.1%. For automatic feature extraction as well as human-like accuracy, researchers are now inclined to neural network architecture [24, 25]. The authors of [26] expanded the image samples on the Bangla Lekha-isolated

character dataset and tested their work on the CNN model with 91.81% accuracy on the alphabets on the base dataset and accuracy of 95.25% after expanding the dataset to 200,000 images using data augmentation techniques such as rotation, zoom, shear, position shifting, shear, etc.

**2.1. Applications of Handwritten Character Recognition.** Handwritten character recognition is one of the major applications of visual document analysis: sorting or reading PIN/ZIP from postal letters, bank check amounts, extracting data from any application form, OCR for blind people, playing a vital role in digital libraries by entering the textual information present in an image into digital formats, helping a great deal to preserve historical documents, and many more. The list below includes some real character recognition system models [27–31].

Google's neural machine translation (NMT) is an end-to-end learning approach for automated translation. NMTs are well known for requiring a high computational cost for both training and making translation inferences. NMT systems are allegedly not robust enough, especially when input phrases contain rare words, according to a number of authors. In comparison to Google's phrase-based production system, the neural machine translation (GNMT) technology reduces translation errors by an average of 60%. On the WMT'14 English-to-French and German benchmarks, GNMT achieves competitive results that are state-of-the-art. The system's accuracy outperforms all previously published findings when measured using a human side-by-side comparison. In a deep LSTM network with 8 encoder and 8 decoder layers, the system model uses attention links from the decoder network to the encoder as well as residual connections. The words were divided into a small set of typical subword units (known as “wordpieces”) for both input and output in order to better handle unusual terms [27].

An open-source OCR engine called Tesseract was created between 1984 and 1994 at Hewlett-Packard. Black-on-black text was perhaps the first to be handled so easily by an OCR engine, according to Tesseract. Tesseract assumes that its input is a binary picture with clearly defined polygonal text sections. At this point, blobs are created solely by stacking outlines together. Text blobs are examined for proportional or fixed-pitch text. Character cells immediately chop the fixed pitch. Words in proportional text are separated using both definite and fuzzy spaces. An adaptive classifier receives each suitable word as training data [29].

In the real world, The Deutsche Post, AG, employed a method for sorting letters to recognize handwritten zip codes. A time-delayed neural network (TDNN) classifier was used to identify hand-printed digits after the machine had read the destination address. A different classifier extracted the structure of each digit and compared it to a range of digits [30].

An OCR in Braille for Blind People: this paper discusses the fundamentals of an optical character recognizer (OCR) for the Braille Code, the writing system used by blind people. With funding from the National Organization of Spanish

Blind People, they created this system. Even with an A4 scanner, the OCR can handle sheets larger than the typical A4 [31].

### 3. Methodology

In this section, the proposed methodology for handwritten character and numeral recognition is provided. In particular, to make it simple and easy to clarify, this section is divided into two subsections: CNN architecture and datasets for Odia language.

**3.1. CNN Architectures.** The CNN algorithm is the most well-known and widely used in the field of deep learning. CNN has a distinct advantage over its predecessors in that it discovers important features without the requirement for human intervention. Computer vision, audio processing, and facial identification are just a few of the applications that CNNs have been used for. Similar to a traditional neural network, the structure of CNNs is also inspired by neurons in human and animal brains. This typical CNN, similar to a multilayer perceptron (MLP), includes numerous convolution layers preceding subsampling (pooling) layers, followed by fully connected (FC) layers.

For the handwritten character recognition of the Odia language, we have implemented five different CNN models. The architecture of a deep learning model can be thought of as its layers. Different types of layers can be employed in the models. Each of these layers has its own significance based on its characteristics. All of the CNN architectures we have implemented here have two convolutional layers followed by one hidden dense layer and another output layer. Extraction of features from images can be done by the convolutional layer, which is the first layer of the CNN architecture. Because pixels are only related to neighbouring and close pixels, convolution preserves the relationship between distinct regions of an image, and it is the process of shrinking an image by filtering it with a smaller filter.

In CNN, pooling layers are frequently added following each convolution layer, which minimises the spatial size of the feature maps. This is another method for reducing overfitting. The pooling method is employed by selecting the maximum, average, or total values inside these pixels. Maximum pooling is one of the most commonly used pooling algorithms, and we employ it after each convolution step in our submitted work.

The dense layer's neurons are all coupled to the neurons in the layer before it. Dense layers are employed in handwritten character recognition to identify images based on the output of the convolutional layer. Each layer of the neural network's neurons computes the input's weighted average and passes it through a nonlinear function, an important part of a neural network's architecture called an activation function. Commonly used activation functions include sigmoid, tanh, step function, linear function, exponential linear unit, ReLU, and leaky ReLU. The rectified linear unit activation function, or ReLU, will produce the same output as the input if the input is positive; otherwise, it will output

zero and is shown in equation (1). We picked ReLU as the default activation function in all five of our constructed CNN models since it is easier to train and provides better results more often.

$$f(x) = \begin{cases} 0, & \text{if } x \leq 0, \\ x, & \text{if } x > 0. \end{cases} \quad (1)$$

Batch normalization is used after each layer as it makes the architecture faster and more stable through normalization of the layers' inputs by recentering and rescaling.

The architectural specifications of five different proposed CNN models applied to the Odia character dataset [32] are represented in Table 1. The number of features in the convolutional layers and the number of units in the dense layer change from one architecture to the next. For all the models, the shape of the input layer is  $28 \times 28 \times 1$ , and final layer is the output layer. Bayesian optimization [33] is used to find the optimal value for the number of features, the number of units, and the learning rate. For the loss function, categorical cross-entropy has been chosen, and the Adam optimizer is used as the optimizer in all the models. Categorical cross-entropy is basically used as a loss function in multiclass classification tasks. This is applicable when there are multiple categories present and the system or model has to choose only one from them. Adam is a wider optimization technique that is used to iteratively adjust network weights based on training data. This optimization method is very efficient and consumes very little memory when dealing with a model with a lot of data or parameters. The layers from 1–6 are mainly used for extracting the features of the input image. Layer 7 flattens the image. Layers 8, 9, and the output layer classify the input image based on the features extracted by the previous layers.

**3.2. Datasets of Odia Language.** Odia, an Indian language, is mostly spoken in the state of Odisha in India (formerly known as Orissa). Native speakers make up to 82% of the population in Odisha, while Odia is also used in parts of Indian states such as Chhattisgarh, Jharkhand, and West Bengal. Due to the roundish structure of Odia letters and the fact that handwriting styles differ from person to person, it is a challenging task for researchers to get human-like classification accuracy. To design a machine learning model, a standard dataset is needed to validate the algorithm. For our research work, we have used the Odia character dataset [32], prepared at NIT Rourkela (NITROHCS v1.0) and the Odia numeral database [34], prepared at ISI, Kolkata. These databases are popular and mostly used as a benchmark database for the research community interested in handwritten digit or character recognition experiments for Odia language.

**3.2.1. NITROHCS v1.0 Database of Handwritten Odia Characters.** This database contains 47 classes of handwritten characters with 320 images in each class, i.e., 15,040 samples in total. The dataset consists of samples that were collected from a total of 160 people from different age groups. At different times, each person has contributed to the samples

twice. The sample characters of the character database are shown in Figure 1.

### 3.2.2. ISI Image Database of Handwritten Odia Numerals.

There are 5000 sample images collected from 356 people in this database of handwritten Odia numerals, which consists of 10 classes. A total of 105 emails and 166 job applications were used to create the database. The entire dataset was already divided into a training set with 4970 samples and a test set with 1000 samples. The sample characters of the numeral datasets are shown in Figure 2.

## 4. Data Augmentation

Data augmentation refers to techniques for enhancing the quantity of data that is accessible by including additional copies of existing data that have been minimally modified or by generating new artificial data from existing data. Deep learning models, which can learn characteristics with multiple layers of abstraction from data, have recently changed the state-of-the-art in many fields. The training of high-dimensional deep learning models like CNN requires the addition of additional data [35]. However, because so many parameters must be learned by these deep learning models, these approaches are prone to overfitting. Larger datasets could serve as regularizers and provide stronger models. But, collecting and manually assigning labels to handwritten images can be a time-consuming and expensive process. As a result, users frequently need to use artificial data augmentation while using datasets with fewer images. In this work, we employ a fully convolutional neural network that can perform at the cutting edge and look into the advantages of adding augmented image samples to the training set that are produced by nonlinearly transforming handwritten images. The data augmentation method involves applying random transformations to the initial training data in order to create new observations by rotating, translating, etc., to the existing ones. Image augmentation is a common activity in medical imaging procedures, including the processing of magnetic resonance images (MRI), X-ray computed tomography (CT), and positron emission tomography (PET) [36–38].

All the samples in both datasets have been augmented for each of the augmentation techniques, and the same split has been used each time as it is used in the case of a normal dataset. The enhanced database size after applying various augmentation strategies is shown in Table 2.

**4.1. Affine Transformations.** Applying mathematical computations to each point, line, and plane of an object to create a new one is known as an “affine transformation,” as a result, the collinearity between points will be preserved [39]. The set of operations providing linear transformations includes translation, rotation, and scaling, and these affine transformations can be applied to an image to expand the dataset. We consider a 2-D image  $I(x, y)$  and a point  $P(x, y)$  and then the point  $P'(ax + by + c, mx + ny + o)$  affine transformed of point  $P(x, y)$ , where  $a, b, c, m, n, o$  are scalar values.

TABLE 1: Architectural specification of the proposed five CNN models M1, M2, M3, M4, and M5.

Layer		Feature map					Size					Kernel size	Stride	Activation			
		Model no.					Model no.										
		1	2	3	4	5	1	2	3	4	5						
Input	Image	1	1	1	1	1						—	—	—			
1	Convolution	45	45	25	5	20						5	1	ReLU			
2	Max pooling	45	45	25	5	20						2	2	—			
3	Batch normalization	45	45	25	5	20						2	-	—			
4	Convolution	75	50	75	75	70						5	1	ReLU			
5	Max pooling	75	50	75	75	70						2	2	—			
6	Batch normalization	75	50	75	75	70						2	—	—			
7	Flatten	—	—	—	—	—	1200	800	1200	1200	1120	—	—	—			
8	Dense	—	—	—	—	—	300	200	300	300	300	—	—	ReLU			
9	Batch normalization	—	—	—	—	—	300	200	300	300	300	—	—	—			
Output		—	—	—	—	—	47	47	47	47	47	—	—	Softmax			



FIGURE 1: Sample characters of NITROHCS v1.0 database.

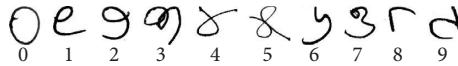


FIGURE 2: Sample numerals of ISI image database.

TABLE 2: Database size after augmentation and splitting information.

Augmentation method	NITROHCS v1.0 handwritten database for Odia character				ISI image database of handwritten oriya numerals			
	Original database		Expanded database		Original database		Expanded database	
	Training	Testing	Training	Testing	Training	Testing	Training	Testing
Translation	10528	4512	94752	4512	4970	1000	44730	1000
Rotation	10528	4512	31584	4512	4970	1000	14910	1000
Scaling	10528	4512	21056	4512	4970	1000	9940	1000
Elastic deformation	10528	4512	21056	4512	4970	1000	9940	1000
Gaussian noise	10528	4512	21056	4512	4970	1000	9940	1000
Color inversion	10528	4512	21056	4512	4970	1000	9940	1000

**4.1.1. Translation.** Translation makes the image move along either the X or Y direction (or both) without changing the shape or angle.  $P' = (x', y')$  is the transformed point of  $P = (x, y)$  and the values of  $x'$  and  $y'$  are given in equations (2) and (3). Figure 3 shows the translation process and some sample translated images of

ISI image database. The parameter values of  $c$  and  $o$  decide the direction of translation.

$$x' = x + c, \quad (2)$$

$$y' = y + o. \quad (3)$$

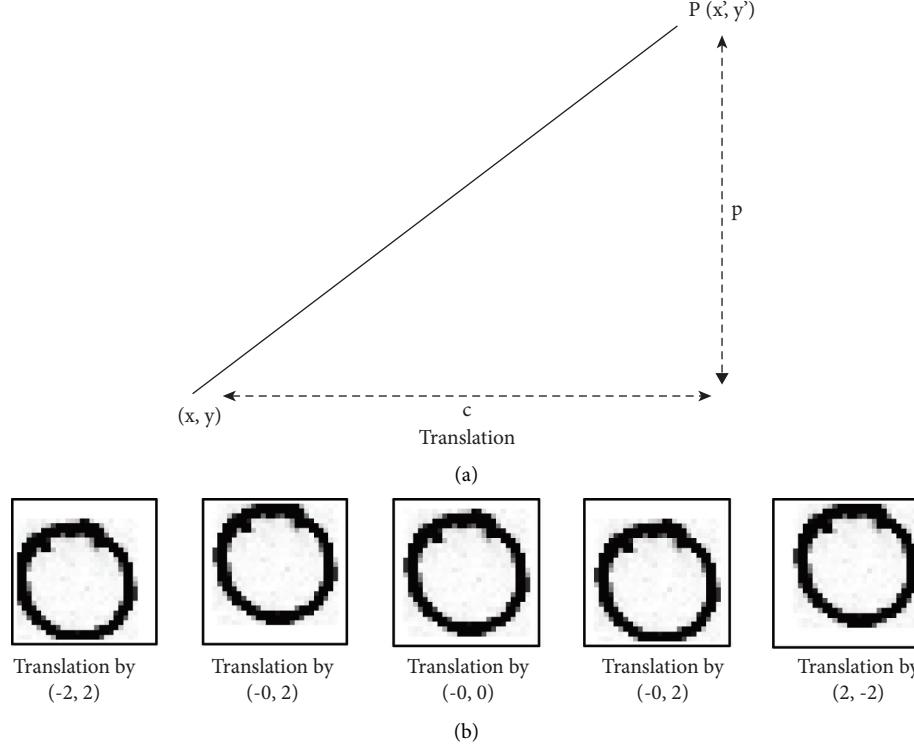


FIGURE 3: (a) The process of translation technique. (b) Example of translated images after applying translation by different  $(x, y)$  values.

TABLE 3: Effect of translation on performance.  $t$  is the parameter describing the amount of translation.

Model	NITROHCSv1.0 Odia character database				ISI image Odia numeral database			
	Original	$t = 1$	$t = 2$	$t = 3$	Original	$t = 1$	$t = 2$	$t = 3$
M1	97.76	98.58	98.27	97.60	97.00	97.89	98.60	98.10
M2	97.65	98.44	98.38	96.85	97.10	98.40	98.50	97.79
M3	97.76	98.51	98.49	96.98	97.39	97.69	98.10	98.00
M4	97.43	98.38	97.80	94.68	97.20	98.10	98.00	97.79
M5	97.83	98.55	98.27	96.63	97.10	97.89	98.50	96.79

Our assumption is that the images have a white background beyond their boundary and are appropriately translated. Such a technique is quite helpful, as most objects can be located anywhere in the image. This ensures that the convolutional neural network looks everywhere in the image. We have restricted only translation to small values because more translation leads to significant portions of characters being removed from the image, which proves detrimental to the performance of the CNN architectures. In the translation technique, for a parameter  $t$ , each of the images in the datasets is translated by  $-t, 0$  and  $t$  pixels in the X direction and by  $-t, 0$  and  $t$  pixels in the Y direction, which increases the size of the dataset by nine times. The process of rotation and sample translated images are shown in Figures 3(a) and 3(b). The images in the datasets are translated by  $-t, 0, t$  pixels in the X and Y directions. From Table 3, it is clear from the experiment that the models achieve better performance on the translated dataset than on the original dataset.

**4.1.2. Rotation.** Rotation involves turning an image along its centre in a clockwise and anticlockwise direction by some randomized number of degrees.  $P' = (x', y')$  is the transformed point of  $P = (x, y)$  after rotation, and the values of  $x'$  and  $y'$  are given in the following equations:

$$x' = x\cos\theta - y\sin\theta, \quad (4)$$

$$y' = x\sin\theta + y\cos\theta. \quad (5)$$

Rotation involves turning an image along its centre in a clockwise or anticlockwise direction by some degrees. Naturally, by locating and extracting a character from the whole image, it is possible to get images even after they are slightly rotated. To make the CNNs robust to such changes, we rotate the images in the dataset by small angles. The image is rotated by  $-t, 0,$  and  $t$  degrees for a parameter  $t$ , which increases the size of the dataset by a factor of three, and sample images are shown in Figures 4(a) and 4(b).

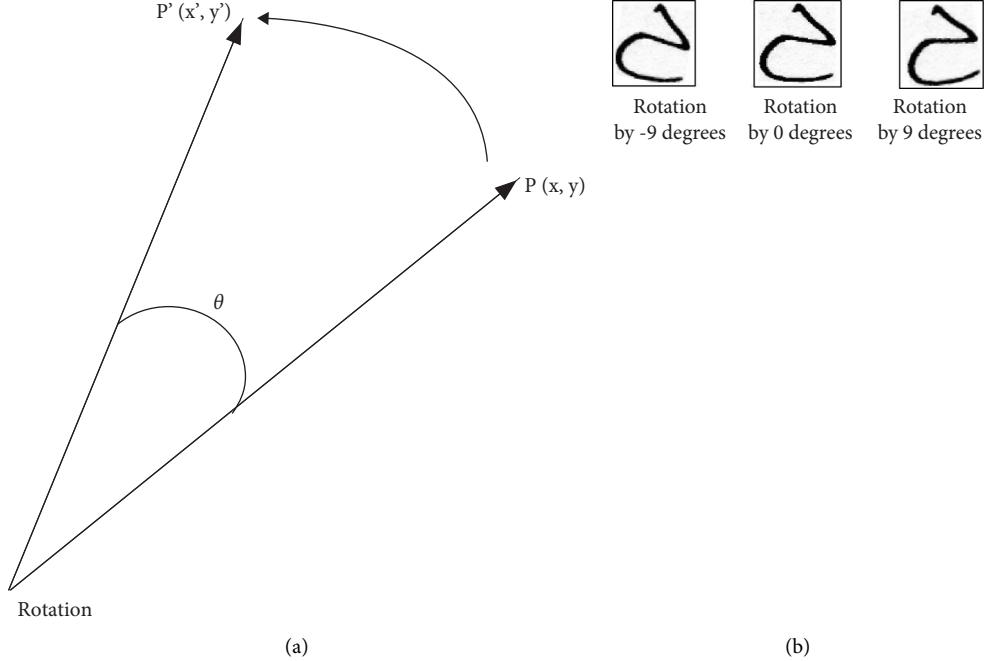


FIGURE 4: (a) Process of rotation technique. (b) Example of rotated images of the number nine.

TABLE 4: Effect of rotation on performance, where  $r$  is the angle in degrees describing the amount of rotation.

Model	NITROHCSv1.0 Odia character database				ISI image Odia numeral database			
	Original	$r=2$	$r=5$	$r=9$	Original	$r=2$	$r=5$	$r=9$
M1	97.76	98.02	97.98	97.98	97.00	97.60	97.60	98.00
M2	97.65	98.24	98.18	98.07	97.10	97.69	97.69	97.79
M3	97.76	97.93	98.11	98.44	97.39	97.60	97.69	97.79
M4	97.43	97.98	97.82	97.69	97.20	97.29	97.10	97.29
M5	97.83	98.13	98.38	98.13	97.10	97.50	97.60	96.69

Rotation involves turning an image along its centre in a clockwise or anticlockwise direction by some degrees. For a parameter  $r$ , all the images are rotated by  $-r$ , 0, and  $r$  degrees. We rotated the images for  $r$ , which ranges from 1 to 10 degrees, and for  $r=2, 5, 9$ , the maximum validation accuracy is achieved and is displayed in Table 4.

**4.1.3. Scaling.** Scaling involves stretching, compressing, or resizing the original image. The scaling point  $P' = (ax, ny)$  of the original point  $P(x, y)$  and the scaling process and sample images after scaling are shown in Figure 5.

Scaling involves resizing the original image. Here, we scale down the original image but add extra white pixels around it to keep the dimension of the resultant image unchanged. For a parameter  $s$ , we reduce the number of rows and the number of columns by  $s$ , which is shown in Figures 5(a) and 5(b). We scale the image by reducing the size of the character without changing the image dimension by a parameter  $s$ , where  $s$  is the parameter describing the amount of reduction. The performance of the scaling operation is shown in Table 5.

**4.2. Elastic Deformation.** Elastic transformation was first introduced in [3]. This paper puts forward that the distribution has invariance not only with respect to elastic deformations, which result from the uncontrolled oscillations of hands damped by inertia, but also with respect to affine transformations. This paper showed that the elastic transformation improved the performance of CNN on the MNIST dataset. We postulate that the same is true for the NITROHCS Odia Character Dataset, ISI, and Kolkata Odia Numeral Dataset, and Figure 6 shows some example images of elastic deformation. From Table 6, it is clear that elastic deformation gives a considerable improvement in performance.

**4.3. Gaussian Noise.** Gaussian noise is statistical noise whose probability density function (PDF) is similar to the normal distribution. The generated noise is then added to the image, which disturbs the gray values present in the digital image. The PDF or normalized histogram of a Gaussian random gray variable “ $x$ ” is given in the following equations:

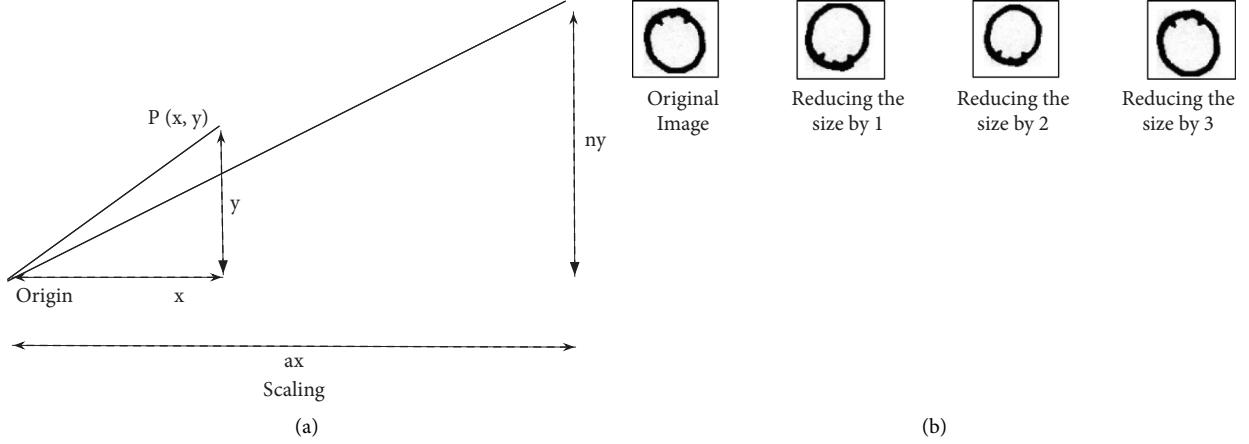


FIGURE 5: (a) Process of scaling technique. (b) Example of original and scaled images.

TABLE 5: Effect of reducing size by scaling on performance.  $s$  is the parameter describing the amount of reduction.

Model	NITROHCSv1.0 Odia character database				ISI image Odia numeral database			
	Original	$s = 1$	$s = 2$	$s = 3$	Original	$s = 1$	$s = 2$	$s = 3$
M1	97.76	98.02	98.05	97.80	97.00	97.29	97.39	97.29
M2	97.65	97.78	97.87	98.18	97.10	97.29	97.60	97.60
M3	97.76	97.85	98.29	97.96	97.39	97.39	97.50	97.50
M4	97.43	97.96	97.76	97.70	97.20	97.60	97.10	97.10
M5	97.83	97.87	98.05	97.87	97.10	97.79	97.69	97.69

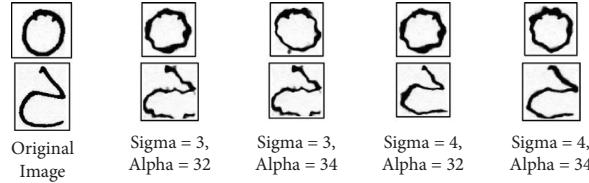


FIGURE 6: Example of elastic deformation.

TABLE 6: Effect of elastic deformation on performance.  $s$  and  $a$  are sigma and alpha, the parameters of elastic deformation.

Model	NITROHCSv1.0 Odia character database				ISI image Odia numeral database			
	Original	$s = 4, a = 34$	$s = 4, a = 32$	$s = 3, a = 32$	Original	$s = 4, a = 34$	$s = 4, a = 32$	$s = 3, a = 32$
M1	97.76	97.78	98.14	97.69	97.10	97.29	97.20	97.10
M2	97.65	97.51	98.02	97.98	96.89	97.60	97.79	96.89
M3	97.76	97.73	98.87	97.96	96.79	97.20	97.39	96.79
M4	97.43	97.56	97.78	97.45	97.20	97.50	97.10	97.20
M5	97.83	97.98	98.89	98.04	97.20	97.69	97.50	97.20

$$p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(x-\mu)^2/2\sigma^2}, \quad (6)$$

$$\int_{-\infty}^{\infty} p(x) dx = 1, \quad (7)$$

where  $\sigma$  is the standard deviation and  $\mu$  is mean. It causes an increase in the size of the dataset by a factor of 2, and sample images after applying Gaussian noise are shown in Figure 7. From Table 7, it is observed that this transformation gives a slight improvement in performance by varying the sigma value.

**4.4. Color Inversion.** Color inversion inverts the color of each pixel. For example, a black character on a white background changes into a white character on a black character. This helps in doubling the size of the dataset, and sample images are shown in Figure 8. From Table 8, it is clear the color inversion gives a considerable improvement in performance.

## 5. Results and Discussion

This section contains a number of simulation results that characterize the performance of the proposed character and numeral recognition algorithms in various benchmark

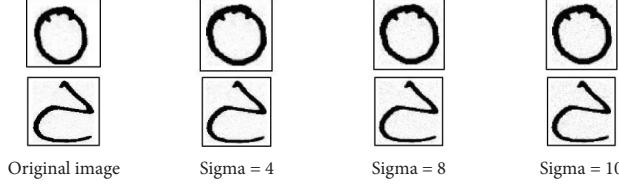


FIGURE 7: Example of images applying Gaussian noise.

TABLE 7: Effect of Gaussian noise on performance by varying sigma (i.e., standard deviation).

Model	NITROHCSv1.0 Odia character database				ISI image Odia numeral database			
	Original	Sigma = 4	Sigma = 8	Sigma = 10	Original	Sigma = 4	Sigma = 8	Sigma = 10
M1	97.76	97.96	97.98	98.07	97.00	97.60	97.29	97.60
M2	97.65	97.85	97.91	98.00	97.10	97.10	97.60	97.39
M3	97.76	97.91	98.91	98.09	97.39	97.20	97.29	97.79
M4	97.43	97.82	97.53	97.96	97.20	97.29	97.10	97.60
M5	97.83	98.02	98.80	98.07	97.10	97.39	97.39	97.29

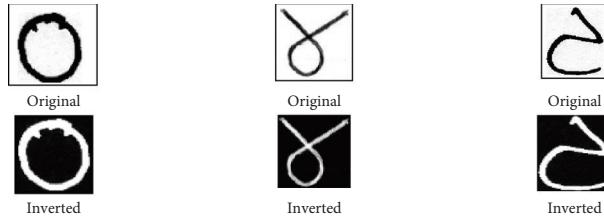


FIGURE 8: Example of color inversion.

TABLE 8: Effect of color inversion on performance.

Model	NITROHCSv1.0 Odia character database		ISI image Odia numeral database	
	Original	Inverted	Original	Inverted
M1	97.76	98.58	97.00	98.00
M2	97.65	98.44	97.10	97.50
M3	97.76	98.51	97.39	97.50
M4	97.43	98.38	97.20	97.60
M5	97.83	98.55	97.10	97.79

datasets. We first compare the baseline model with the regularized CNN model. Furthermore, we compare the effects of different augmentation techniques on the performance of the proposed five different CNN models for character and numeral recognition, and we compare the proposed method with the state-of-the-art recognition methods.

**5.1. Baseline vs. Regularized Model.** All experiments were carried out on the Google Colab in a GPU environment, and the experimental results reported in this section are for the designed five baseline models M1, M2, M3, M4, and M5. The NITROHCS v1.0 character database did not have separate training and testing examples. Hence, a 70%-30% split was used to obtain training and testing examples, whereas the ISI image numeral database was split into training and testing sets, and the same split has been used without any changes. It is observed from Figure 9 that the baseline model achieves a training accuracy of 100% after 10 epochs, but its validation

accuracy stagnates around 97%. This applies to all five models on both the character and numeral datasets. This is a sign that the models are overfitting the data. To avoid overfitting of the models, L2 regularization and spatial dropout were added to the models [40]. It can be observed that in the regularized models, the gap between the training accuracy and validation accuracy has reduced in all the models. From Table 9, it is clear that the maximum validation accuracy achieved by the models has increased after the application of regularization.

**5.2. Effect of Data Augmentation on Performance.** In our experiment, we applied different data augmentation techniques, i.e., translation, rotation, scaling, etc. The maximum validation accuracy for two different character and numeral datasets is compared for five different models.

Table 2 shows the enhanced database size after applying various augmentation strategies. Tables 10 and 11 compare the performance of various available handwritten character

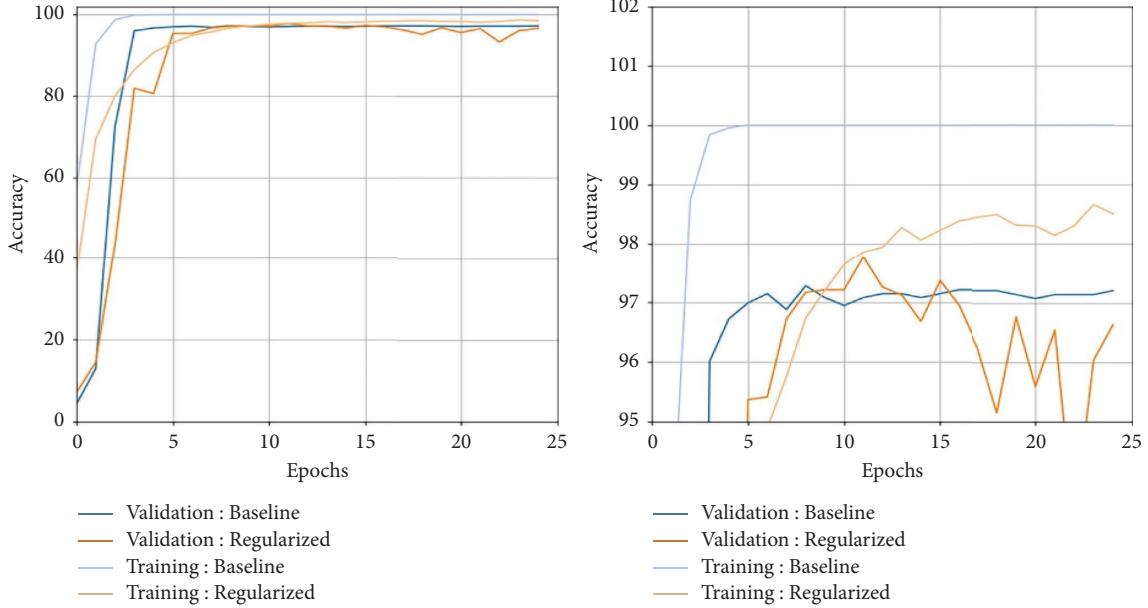


FIGURE 9: Baseline vs. regularized performance of model M1 on Odia character dataset.

TABLE 9: Baseline models vs. regularized models in the Odia character database and the Odia numeral database.

Model	Maximum validation accuracy			
	NITROHCSv1.0 Odia character database		ISI image Odia numeral database	
	Baseline	Regularized	Baseline	Regularized
M1	97.27	97.76	97.00	97.00
M2	97.49	97.65	96.60	97.10
M3	97.42	97.76	96.49	97.39
M4	96.88	97.43	95.99	97.20
M5	97.16	97.83	96.60	97.10

TABLE 10: Performance comparison on handwritten ISI image and Odia numeral databases.

Database	Work reference	Features	Recognition classifier	Recognition accuracy (%)
ISI image database of handwritten oriya numerals	[11]	Binary external symmetry axis constellation	Random forest	98.44
	[41]	Scalar and stroke information	HMM	90.50
	[42]	LU decomposition of matrix factors	Backpropagation ANN	85.30
	[14]	Gradient, curvature	Low complexity neural network	98 (gradient) 94 (curvature)
		Slantlet transform, stockwell transform and gabor	k-NN	95.04 (slantlet) 98.08 (stockwell)
	[16]	stockwell transform and gabor		
	Proposed work (without augmentation)	Features obtained from convolutional layer	CNN	97.39
	Translation Rotation Scaling Elastic deformation Noise Color inversion			98.60 98.00 97.79
	Proposed work (with augmentation)	Features obtained from convolutional layer	CNN	97.69
				97.60 98.00

TABLE 11: Performance comparison on handwritten NITROHCS v1.0 Odia character databases.

Database	Work reference	Features	Recognition classifier	Recognition accuracy (%)	
NITROHCSv1.0 handwritten database for Odia character	[43]	Statistical features as euclidian distance, Hamilton distance	RBFNN	98.30	
	[44]	LeNet-5 CNN	HopField NN	95.00	
	Proposed work (without augmentation)	Features obtained from convolutional layer	CNN	97.76	
	Translation			98.58	
	Rotation			98.24	
	Scaling			98.18	
	Proposed work (with augmentation)	Elastic deformation	Features obtained from convolutional layer	CNN	98.89
	Noise			98.91	
	Color inversion			98.58	

TABLE 12: Performance comparison of data augmentation techniques.

Reference	Dataset	Number of classes	Classifier	Accuracy without augmentation	Accuracy with augmentation (%)
[26]	BanglaLekha-isolated	50	CNN	91.81%	95.25
[21]	ISI image handwritten Oriya numerals	10	GAN	96.40%	97.30
[21]	ISI image handwritten Bangla numerals	10	GAN	98.35%	98.55
[21]	ISI image handwritten Devanagari numerals	10	GAN	98.48%	98.56
Proposed work	NITROHCSv1.0 Odia characters	47	CNN	97.76%	98.91
Proposed work	ISI image handwritten Oriya numerals	10	CNN	97.39	98.60

TABLE 13: A comparative study for multilingual handwritten character and numeral recognition.

Reference	Language (dataset)	Methods	Digit recognition accuracy	Character recognition accuracy
[24]	CMATERdb-Bangla	DenseNet	99.13	98.31
[25]	CMATERdb-Bangla	Modified ResNet-18	—	95.10
[23]	Self-prepared database-Bangla	Celled projection (CP) + k-NN	94.12	—
[22]	CMATERdb-Bangla	LRF, HOG and diagonal feature + SVM	—	88.73
[45]	DHCD-Devanagari	Deep CNN	—	98.47
[46]	Self-prepared database-Devanagari	Chain code histogram and moment invariant features + MLP	—	98.03
[47]	Self prepared database-Devanagari	Curvelet transform and the character geometry + k-NN	—	93.8
[48]	ISI Kolkata numeral database	CNN with genetic algorithm	96.41	—
[49]	CMATERdb-Telugu	Discrete wavelet transform (DWT), projection profile (PP) and singular value decomposition (SVD) + k-nearest neighbor (k-NN) and support vector machine (SVM)	—	95.47 (SVM on DWT features)
[50]	Telugu database	Binary external symmetry axis constellation features + quadratic discriminant classifier and SVM	—	80.6 (SVM) 87.6 (QDA)
[51]	MNIST, CMATERdb, ISI, Gujarati and Punjabi badabase	CNN	96.23	—

recognition techniques for the Odia language on the NITROHCS v1.0 character and ISI image numeral datasets.

The data augmentation techniques to expand the dataset are popular in daily life applications like face, speech, or text recognition and classification for different languages, and they also play an important role in medical imaging field also. But unfortunately, very limited contribution from different data augmentation techniques for Odia handwritten character recognition were found. The performance comparison of data augmentation techniques on different datasets is shown in Table 12.

A comparison among the recognition or classification accuracy of handwritten characters and numerals belonging to different Indian languages is shown in Table 13.

## 6. Conclusions

In this work, five variants of 2-layer CNNs are used for handwritten character recognition of Odia characters and numerals. The accuracy of five different baseline as well as regularized model is figured out. After testing the effectiveness of various data augmentation techniques on the Odia characters using the standard character and numeral datasets and providing the augmented dataset as input to five different CNN architectures, we conclude that when the original dataset is either color inverted or Gaussian noise is applied to the dataset, the models produce better accuracy, i.e., 98.91%, than the normal dataset. Other techniques such as translation and rotation also showed slight improvement in accuracy.

## Data Availability

The OHCSv1.0 data used to support the findings of this study have been deposited in the NIT, Rourkela, India, repository (DOI: 10.1109/NCVPRIPG.2015.7490020). The ISI image Odia numerals data used to support the findings of this study have been deposited in the ISI Kolkata, India repository (DOI: 10.1109/ICDAR.2005.84).

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## References

- [1] Y. Bengio, “Learning deep architectures for AI,” 2009, <http://www.iro.umontreal.ca/bengioy>.
- [2] G. E. Hinton, “Learning multiple layers of representation,” *Trends in Cognitive Sciences*, vol. 11, no. 10, pp. 428–434, 2007.
- [3] P. Y. Simard, D. Steinkraus, and J. C. Platt, “Best practices for convolutional neural networks applied to visual document analysis,” in *Proceedings of the 7th International Conference on Document Analysis and Recognition*, 2003, Edinburgh, UK, August 2003.
- [4] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” 2012, <http://code.google.com/p/cuda-convnet/>.
- [5] Q. v. Le, W. Y. Zou, S. Y. Yeung, and A. Y. Ng, “Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis,” 2011, <http://ai.stanford.edu/>.
- [6] K. S. Dash, N. B. Puhan, and G. Panda, “Odia character recognition: a directional review,” *Artificial Intelligence Review*, vol. 48, no. 4, pp. 473–497, Dec. 2017.
- [7] M. Das, M. Panda, and S. Dash, “A comparative analysis of machine learning techniques for Odia character recognition,” *Machine Learning Applications*, pp. 65–90, Apr. 2020.
- [8] Y. Lecun, L. E. Bottou, Y. Bengio, and P. H. Abstract, “Gradient-based learning applied to document recognition,” *Proceedings of the IEEE*, vol. 86, 1998.
- [9] B. B. Chaudhuri, U. Pal, and M. Mitra, “Automatic recognition of printed Oriya script,” *Sadhana*, vol. 27, no. 1, pp. 23–34, 2002.
- [10] R. Kumar Mohapatra, B. Majhi, and S. Kumar Jena, “Printed Odia digit recognition using finite automaton,” *Proceedings of 3rd International Conference on Advanced Computing, Networking and Informatics*, Springer, Berlin/Heidelberg, Germany, 2015.
- [11] K. S. Dash, N. B. Puhan, and G. Panda, “BESAC: binary External Symmetry Axis Constellation for unconstrained handwritten character recognition,” *Pattern Recognition Letters*, vol. 83, pp. 413–422, 2016.
- [12] M. Das and M. Panda, “An ensemble method of feature selection and classification of Odia characters,” in *Proceedings of the 1st Odisha International Conference on Electrical Power Engineering, Communication and Computing Technology, ODICON 2021*, Bhuba, Odisha, January 2021.
- [13] M. Husnain, M. M. Saad Missen, S. Mumtaz et al., “Recognition of Urdu handwritten characters using convolutional neural network,” *Applied Sciences*, vol. 9, no. 13, p. 2758, 2019.
- [14] B. Majhi, J. Satpathy, and M. Rout, “Efficient recognition of Odia numerals using low complexity neural classifier,” in *Proceedings of the 2011 International Conference on Energy, Automation and Signal ICEAS - 2011*, pp. 140–143, Bhubaneswar, India, December 2011.
- [15] D. Khedidja and M. Hayet, “Multiple classifiers and invariant features extraction for digit recognition,” *International Journal of Computer and Electrical Engineering*, vol. 11, no. 1, pp. 41–52, 2019.
- [16] K. S. Dash, N. B. Puhan, and G. Panda, “On extraction of features for handwritten Odia numeral recognition in transformed domain,” in *Proceedings of the ICAPR 2015 - 2015 8th International Conference on Advances in Pattern Recognition*, Kolkata, India, January 2015.
- [17] P. K. Sarangi and P. Ahmed, “Recognition of handwritten Odia numerals using artificial intelligence techniques,” *International Journal of Computer Science and Applications*, vol. 2, 2013.
- [18] 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, 2020.
- [19] R. B. S. D. S. Jana, “Handwritten digit recognition using convolutional neural networks,” in *Deep Learning, Bhattacharyya Siddhartha, Snasel Vaclav, Hassanien Aboul Ella, Saha Satadal, B. K. Tripathy, Ed.*, pp. 51–68, De Gruyter, Berlin, Germany, 2020.
- [20] M. A. Chhajro, “Handwritten Urdu character recognition via images using different machine learning and deep learning techniques,” *Indian Journal of Science and Technology*, vol. 13, no. 17, pp. 1746–1754, May 2020.
- [21] G. Jha and H. Cecotti, “Data augmentation for handwritten digit recognition using generative adversarial networks,”

- Multimedia Tools and Applications*, vol. 79, no. 47–48, pp. 35055–35068, 2020.
- [22] M. R. Kibria, A. Ahmed, Z. Firdawsi, and M. A. Yousuf, “Bangla compound character recognition using support vector machine (SVM) on advanced feature sets,” in *Proceedings of the 2020 IEEE Region 10 Symposium, TENSYMP 2020*, pp. 965–968, Dhaka, Bangladesh, June 2020.
  - [23] M. Z. Hossain, M. A. Amin, and H. Yan, “Rapid feature extraction for Bangla handwritten digit recognition,” in *Proceedings of the 2011 International Conference on Machine Learning and Cybernetics*, vol. 4, pp. 1832–1837, Guilin, China, July 2011.
  - [24] M. Adnan, F. Rahman, M. Imrul, N. Al, and S. Shabnam, “Handwritten Bangla character recognition using inception convolutional neural network,” *International Journal of Computer Application*, vol. 181, no. 17, pp. 48–59, 2018.
  - [25] M. Al Rabbani Alif, S. Ahmed, and M. A. Hasan, “Isolated Bangla handwritten character recognition with convolutional neural network,” in *Proceedings of the 20th International Conference of Computer and Information Technology, ICCIT 2017*, pp. 1–6, Dhaka, Bangladesh, December 2017.
  - [26] P. Chakraborty, A. Islam, M. Abu Yousuf, R. Agarwal, and T. Choudhury, “Bangla handwritten character recognition using convolutional neural network,” *Machine Intelligence and Data Science Applications*, vol. 132, pp. 721–731, 2022.
  - [27] Y. Wu, “Google’s neural machine translation system: bridging the gap between human and machine translation,” pp. 1–23, 2016, <https://arxiv.org/abs/1609.08144>.
  - [28] C. E. Palazzi, S. Ferretti, S. Cacciaguerra, and M. Roccetti, “Interactivity-loss avoidance in event delivery synchronization for mirrored game architectures,” *IEEE Transactions on Multimedia*, vol. 8, no. 4, pp. 874–879, 2006.
  - [29] H. Sharifi and A. Shahbahrami, “A comparative study on different license plate recognition algorithms,” *Communications in Computer and Information Science*, vol. 167, pp. 686–691, 2011.
  - [30] M. Pfister, S. Behnke, and R. Rojas, “Recognition of handwritten ZIP codes in a real - world non-standard-letter sorting system,” *Applied Intelligence*, vol. 12, no. 1/2, pp. 95–115, 2000.
  - [31] X. F. Hermida, A. C. Rodriguez, and F. M. Rodriguez, “A BRAILLE O . C . R . FOR BLIND PEOPLE,” in *Proceedings of the 7th International Conference on Signal Processing Applications & Technology*, Shenzhen China, May 2022.
  - [32] R. K. Mohapatra, T. K. Mishra, S. Panda, and B. Majhi, “OHCS: a database for handwritten atomic Odia Character Recognition,” in *Proceedings of the 2015 5th National Conference on Computer Vision, Pattern Recognition, Image Processing and Graphics*, Mandi, India, December 2016.
  - [33] J. Snoek, H. Larochelle, R. P. Adams, and R. P. Adams, “Practical Bayesian optimization of machine learning algorithms,” 2012, <http://nrs.harvard.edu/urn-3:HUL.InstRepos:11708816>.
  - [34] U. Bhattacharya and B. B. Chaudhuri, “Databases for research on recognition of handwritten characters of Indian scripts,” in *Proceedings of the International Conference on Document Analysis and Recognition, ICDAR*, pp. 789–793, Lausanne, Switzerland, December 2005.
  - [35] S. Hauberg, O. Freifeld, A. B. Lindbo Larsen, J. W. Fisher, and L. K. Hansen, “Dreaming more data: class-dependent distributions over diffeomorphisms for learned data augmentation,” in *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics, AISTATS 2016*, pp. 342–350, Cadiz, Spain, May 2016.
  - [36] A. Zhao, G. Balakrishnan, F. Durand, J. v. Guttag, and A. v. Dalca, “Data augmentation using learned transformations for one-shot medical image segmentation,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 8535–8545, Seattle, WA, USA, June 2019.
  - [37] M. M. Thaha, K. P. M. Kumar, B. S. Murugan, S. Dhanasekeran, P. Vijayakarthick, and A. S. Selvi, “Brain tumor segmentation using convolutional neural networks in MRI images,” *Journal of Medical Systems*, vol. 43, no. 9, pp. 294–1251, 2019.
  - [38] A. Oliveira, S. Pereira, and C. A. Silva, “Augmenting data when training a CNN for retinal vessel segmentation: how to warp?” in *Proceedings of the ENBENG 2017 - 5th Portuguese Meeting on Bioengineering*, Coimbra, Portugal, February 2017.
  - [39] J. M. van Verth and L. M. Bishop, *Affine Transformations Essential Mathematics For Games And Interactive Applications*, pp. 163–202, CRC Press, Boca Raton, FA, USA, 2020.
  - [40] M. Nielsen, “Deep learning,” 2022, <http://neuralnetworksanddeeplearning.com/chap6.html>.
  - [41] T. K. Bhowmik, S. K. Parui, U. Bhattacharya, and B. Shaw, “An HMM based recognition scheme for handwritten Oriya numerals,” in *Proceedings of the 9th International Conference on Information Technology (ICIT’06)* 2006, pp. 105–110, Bhubaneswar, India, December 2007.
  - [42] P. K. Sarangi, P. Ahmed, and K. K. Ravulakollu, “Naïve bayes classifier with LU factorization for recognition of handwritten Odia numerals,” *Indian Journal of Science and Technology*, vol. 7, 2014.
  - [43] A. Sethy and P. K. Patra, “Off-line Odia handwritten character recognition: an axis constellation model based research,” *International Journal of Innovative Technology and Exploring Engineering*, vol. 8, no. 2, pp. 788–793, 2019.
  - [44] R. C. Sahoo, S. K. Pradhan, and P. Tanwar, “HopNet based associative memory as FC layer in CNN for Odia character classification,” in *Proceedings of the Confluence 2020 - 10th International Conference on Cloud Computing, Data Science and Engineering*, pp. 178–182, Noida, India, January 2020.
  - [45] S. Acharya, A. K. Pant, and P. K. Gyawali, “Deep learning based large scale handwritten Devanagari character recognition,” in *Proceedings of the SKIMA 2015 - 9th International Conference on Software, Knowledge, Information Management and Applications*, Kathmandu, Nepal, December 2016.
  - [46] S. Arora, D. Bhattacharjee, M. Nasipuri, D. K. Basu, and M. Kundu, “Application of statistical features in handwritten devnagari character recognition,” 2010, <https://arxiv.org/ftp/arxiv/papers/1006/1006.5911.pdf>.
  - [47] B. Singh, A. Mittal, and D. Ghosh, “Evaluation of different feature extractors and classifiers for offline handwritten devnagari character recognition,” *Journal of Pattern Recognition Research*, vol. 6, no. 2, pp. 269–277, 2011.
  - [48] A. Trivedi, S. Srivastava, A. Mishra, A. Shukla, and R. Tiwari, “Hybrid evolutionary approach for Devanagari handwritten numeral recognition using Convolutional Neural Network,” *Procedia Computer Science*, vol. 125, pp. 525–532, 2018.
  - [49] J. Jyothi, K. Manjusha, M. Anand Kumar, and K. P. Soman, “Innovative feature sets for machine learning based Telugu character recognition,” *Indian Journal of Science and Technology*, vol. 8, p. 24, 2015.
  - [50] N. Prameela, P. Anjusha, and R. Karthik, “Off-line Telugu handwritten characters recognition using optical character recognition,” in *Proceedings of the International Conference on Electronics, Communication and Aerospace Technology, ICECA 2017*, pp. 223–226, Coimbatore, India, December 2017.
  - [51] D. Gupta and S. Bag, “CNN-based multilingual handwritten numeral recognition: a fusion-free approach,” *Expert Systems with Applications*, vol. 165, Article ID 113784, 2021.