Computers, Materials & Continua

Tech Science Press

DOI: 10.32604/cmc.2024.049296

ARTICLE

Development of a Lightweight Model for Handwritten Dataset Recognition: Bangladeshi City Names in Bangla Script

Md. Mahbubur Rahman Tusher[1], Fahmid Al Farid[2,*], Md. Al-Hasan[1], Abu Saleh Musa Miah[1], Susmita Roy Rinky[1], Mehedi Hasan Jim[1], Sarina Mansor[2], Md. Abdur Rahim[3] and Hezerul Abdul Karim[2,*]

[1]Department of Computer Science and Engineering, Bangladesh Army University of Science and Technology (BAUST), Saidpur, 5310, Bangladesh

[2]Faculty of Engineering, Multimedia University, Cyberjaya, 63100, Malaysia

[3]Pabna University of Science and Technology, Pabna, 6600, Bangladesh

*Corresponding Authors: Fahmid Al Farid. Email: fahmid.farid@mmu.edu.my; Hezerul Abdul Karim. Email: hezerul@mmu.edu.my

Received: 02 January 2024 Accepted: 30 April 2024 Published: 15 August 2024

ABSTRACT

The context of recognizing handwritten city names, this research addresses the challenges posed by the manual inscription of Bangladeshi city names in the Bangla script. In today's technology-driven era, where precise tools for reading handwritten text are essential, this study focuses on leveraging deep learning to understand the intricacies of Bangla handwriting. The existing dearth of dedicated datasets has impeded the progress of Bangla handwritten city name recognition systems, particularly in critical areas such as postal automation and document processing. Notably, no prior research has specifically targeted the unique needs of Bangla handwritten city name recognition. To bridge this gap, the study collects real-world images from diverse sources to construct a comprehensive dataset for Bangla Hand Written City name recognition. The emphasis on practical data for system training enhances accuracy. The research further conducts a comparative analysis, pitting state-of-the-art (SOTA) deep learning models, including EfficientNetB0, VGG16, ResNet50, DenseNet201, InceptionV3, and Xception, against a custom Convolutional Neural Networks (CNN) model named "Our CNN." The results showcase the superior performance of "Our CNN," with a test accuracy of 99.97% and an outstanding F1 score of 99.95%. These metrics underscore its potential for automating city name recognition, particularly in postal services. The study concludes by highlighting the significance of meticulous dataset curation and the promising outlook for custom CNN architectures. It encourages future research avenues, including dataset expansion, algorithm refinement, exploration of recurrent neural networks and attention mechanisms, real-world deployment of models, and extension to other regional languages and scripts. These recommendations offer exciting possibilities for advancing the field of handwritten recognition technology and hold practical implications for enhancing global postal services.

KEYWORDS

Handwritten recognition; Bangladeshi city names; Bangla handwritten city name; automated postal services

Nomenclature

CNN	Convolutional Neural Networks
OCR	Optical Character Recognition
ReLU	The Rectified Linear Unit
SOTA	State-of-the-Art

1 Introduction

In the ever-evolving landscape of the digital age, the process of converting handwritten documents into machine-readable formats stands as a pillar of modern technological innovation [1]. The critical task of handwritten city name recognition lies at the forefront of this transformative journey. This task is paramount in document analysis and automation, specifically in applications such as postal automation and document digitization [2]. The uses of Handwritten city names are shown in Fig. 1. Recognizing handwritten city names might seem simple, but it's actually crucial for many important tasks. It's the key to making different systems work well and making sure they get things right.

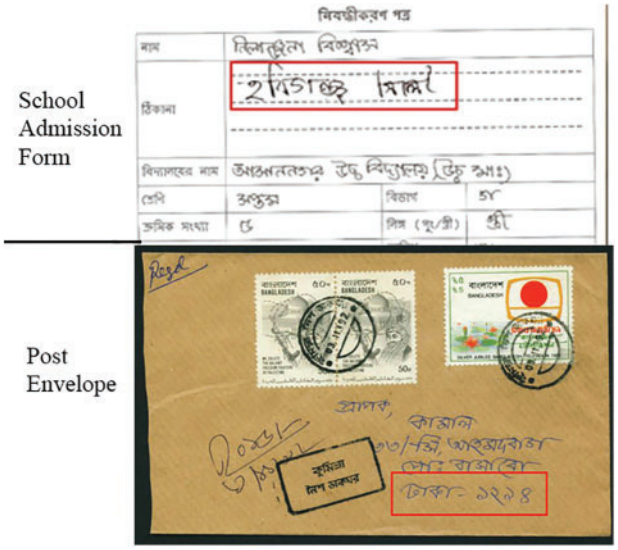


Figure 1: Uses of Bangla handwriting city name

This is important to recognize Bangla handwritten city names, experiencing an oversight more so on digital postal automation and document analysis. For example, the lack of a dedicated framework results in ineffective sorting of mails and difficulties in retrieving information from English city names. Though city names written in English exhibit pretty accurate search results [3], the lack of a robust recognition system for Bangla handwritten text brings variation and poses challenges in handling the linguistic variation that is vitally important to a globalized society, as shown in Fig. 2.

In this situation it is very important to improve digital postal automation and information retrieval accuracy of Bangla handwritten city names, our motivation is driven by. We do so to nurture inclusiveness, accessibility, and efficacy in document analysis and technological innovation beyond the boundaries of regions.

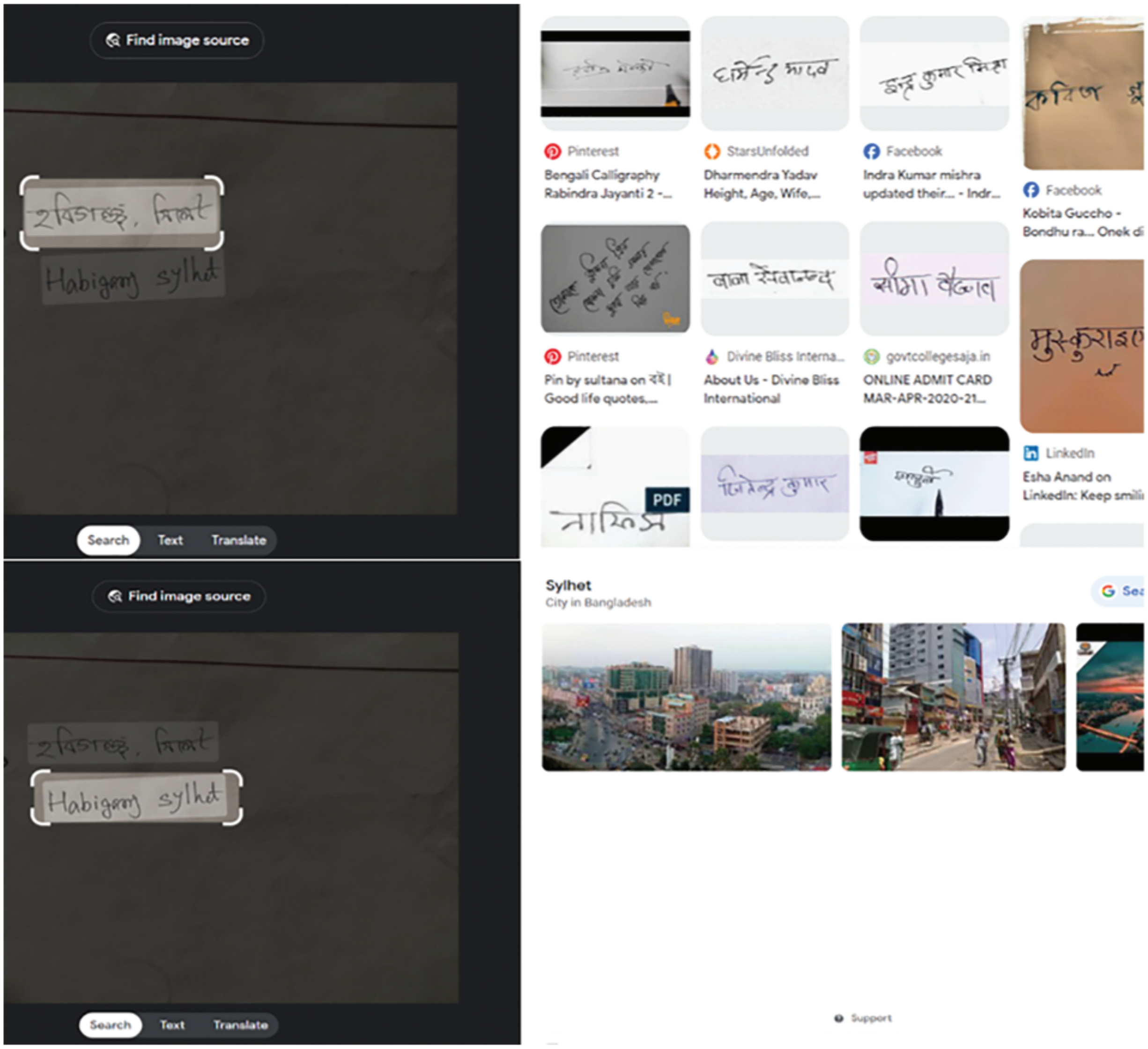

Find image source
হবিগঞ্জ, সিলেট
Habiganj sylhet
Search
Text
Translate
Find image source
Pinterest
Bengali Calligraphy Rabindra Jayanti 2 -...
StarsUnfolded
Dharmendra Yadav Height, Age, Wife,...
Facebook
Indra Kumar mishra updated their... - Indr...
Facebook
Kobita Guccho - Bondhu ra... Onek di
Pinterest
Pin by sultana on বই | Good life quotes,...
Divine Bliss Interna...
About Us - Divine Bliss International
govtcollegesaja.in
ONLINE ADMIT CARD MAR-APR-2020-21...
LinkedIn
Esha Anand on LinkedIn: Keep smili
PDF
Sylhet
City in Bangladesh
Support

Figure 2: An example of search result in google lens with Bangla and English handwritten text

Historically, Optical Character Recognition (OCR) systems have been the stalwart for recognizing printed and handwritten text, significantly improving data processing efficiency [4]. With its capacity to decipher both typed and neatly printed scripts, OCR has been the bedrock for document digitization [5].

Many researchers have been using machine learning algorithms to recognize OCR [6]. The problem of this algorithm is that they face difficulty to handle the large scale dataset. To overcome The problem many researchers, use Deep Learning for English word recognition [7–9] and very few researchers work with Bangla Handwritten [10]. However, as we navigate the intricacies of handwritten text recognition, a critical shift emerges, leading us to the enthralling realm of CNN.

CNN, an integral component of deep learning, has earned its eminence for its remarkable prowess in image recognition tasks [11], making it an ideal choice for handwritten city name recognition. The allure of CNN lies in its unique ability to adapt to complex image patterns and shapes, particularly within the realm of handwritten text [12], which is marked by its intricacy and variability. CNN's flexibility and adaptability enable it to learn and discern the subtle nuances within handwritten city names, making it a formidable contender in the quest for accurate recognition.

The rationale behind choosing CNN over OCR for handwritten city name recognition lies in the nuanced nature of handwritten text. Handwriting, often marked by distinctive and highly varied styles [13], defies the rigid structures of printed or typed text. Unlike OCR, which often relies on predefined character patterns and segmentation, CNN takes a holistic approach [14]. It looks at the entire image, recognizing patterns and structures without explicit segmentation. This adaptability allows CNN to accommodate the myriad ways city names are handwritten, irrespective of style, size, or spacing variations.

The significance of handwritten city name recognition reverberates across many domains, with the field of postal automation serving as an illustrative example [15]. Postal services heavily rely on the capacity to read and process handwritten addresses on envelopes and parcels [16]. The recognition of handwritten city names is, thus, instrumental in streamlining mail sorting and, by extension, revolutionizing the postal sector's overall efficiency. Moreover, this technology extends its influence into other administrative processes, such as bank check reading and insurance claim processing, promising to reshape the landscape of document-related tasks, significantly reducing the margin for error and human intervention [17]. Few research work found in CNN for Bangla Handwritten recognition faces some challenges, the main challenges in Bangla Handwritten Word Recognition are the lack of a well-labeled dataset and the relatively low accuracy of recognition using segmentation-free approaches [8,18]. In this situation it is urgent to develop Bangla Handwritten city name recognition system and the dataset resource. We proposed a lightweight CNN model to recognize Bangla Handwritten city names. We define the proposed CNN model as a lightweight model because we used only a line convolutional layer that needs minimum parameter and computational resources. The main contribution of the paper is given below.

Novelty: Curated comprehensive data from all 64 districts of Bangladesh for city name recognition in Bangla Handwritten.

Methodological Innovation: Designed and implemented a dedicated CNN for proficient Bangladeshi District Name recognition beating state-of-the-art models in accuracy and performance. We define proposed CNN model as a lightweight model, our lightweight model has only 9 CNN layers with 5.002 parameters outperforms SOTA models in Bangladeshi District Name recognition, highlighting efficiency despite its lightweight architecture.

Empirical Validation: Achieved a remarkable recognition accuracy of 99.97% through rigorous experimentation on the newly established dataset. The data and code have been uploaded to the following link https://github.com/tusher100/CIty-Name-Recognition (accessed on 31 December 2023).

The paper's structure comprehensively explores Bangla handwritten city name recognition. It begins with Section 2, a thorough literature review discussing prior research and SOTA models; Section 3 provides a detailed description of the specially curated dataset, followed by Section 4, which elucidates the methodology, including model architectures and training; Section 5 delves into an in-depth analysis of the results, including a comparative evaluation against SOTA models. Finally, Section 6 summarizes the study's conclusions, underlining the dataset's significance and the "Our CNN" model's exceptional performance while considering future research directions.

2 Literature Review

In recent years, significant advancements have been made in handwritten city name recognition, driven by the growing need for efficient postal automation systems and applications in multilingual and multiscript countries like India. A wide range of scripts, including Gurmukhi, Arabic, Bangla, Bengali, Farsi, and Tamil, highlight the importance of this research in the context of postal automation and other applications.

One noteworthy contribution is the introduction of the HWR-Gurmukhi_Postal_1.0 dataset for handwritten city name recognition in Gurmukhi script [19]. This benchmark dataset is crucial for evaluating existing techniques and advancing research in offline handwritten Gurmukhi recognition. The availability of such datasets is fundamental for fostering developments in this specialized field. The IfN/Farsi database, introduced by author Mozaffari et al. [20], provides a resource for handwritten Arabic word recognition with a collection of 7271 binary images. While specific recognition accuracy percentages are not provided, the detailed ground truth information adds value for research and academic purposes. An exploration of CNN and the influence of different learning rates on handwritten district names in Gurmukhi Script reveals valuable insights (Sharma et al. [21]). A maximum validation accuracy of 99% with a learning rate of 0.0001 demonstrates the significance of learning rate optimization in achieving robust and accurate recognition models. These findings offer crucial guidance for enhancing handwritten text recognition systems. The recognition of Arabic handwritten city names is addressed in Mahjoub et al. [22], which introduces novel approaches using Bayesian network classifiers. The Forest Augmented Naïve Bayes (FAN) and Dynamic Bayesian Network (DBN) models exhibit promising recognition rates of up to 83.7%. This research showcases the effectiveness of Bayesian network classifiers in the context of Arabic handwritten word recognition, opening new avenues for accuracy improvement. Nurseitov et al. [17] delve into the underrepresented field of handwritten recognition specific to Cyrillic script, focusing on Kazakh and Russian languages. The study introduces four distinct deep learning models, providing valuable insights into their performance. The Wordbeamsearch model stands out in city name classification with an impressive accuracy of 75.1%. In the context of Handwritten Text Recognition (HTR), the SimpleHTR model excels, achieving a remarkable Character Error Rate (CER) of 1.55% and a Word Error Rate (WER) of 11.09%, filling a significant gap in the existing research. The development of a CNN based model for recognizing handwritten district names in Gurmukhi script showcases a substantial leap in accuracy [23]. Achieving a maximum validation accuracy of 99%, this work eliminates the need for manual feature extraction, offering immense potential for postal automation. Barua et al. [24] presents a holistic approach for recognizing handwritten city names in Bangla script with an accuracy of 90.65%. While showing promise for postal automation, the approach acknowledges areas for improvement, such as feature dimension reduction and addressing misclassification among structurally similar regions. Prasad et al. [25], recognition of Bengali place names as word images are explored using different CNN architectures. This study introduces a dataset of Bengali word images, demonstrating competitive accuracy and robustness in word-level recognition, which is essential for applications like postal automation. The significance of feature selection techniques is highlighted in Kumar et al. [26] for offline handwritten Gurumukhi place name recognition. The CSA feature selection technique, combined with a random forest classifier, achieves an impressive recognition rate of 87.42%.

Ghosh et al. [27] introduce a Memetic Algorithm (MA)-based Wrapper-filter feature selection framework for recognizing handwritten Bangla city names. While specific recognition accuracy is not provided, the paper underscores the importance of feature selection in improving performance. Sahoo et al. [28], a novel shape-context-based 64-dimensional feature vector, is introduced for recognizing handwritten city names in the Bangla script. Although specific accuracy percentages are not given, the paper highlights the effectiveness of this feature vector in achieving higher recognition accuracy. Pal et al. [29], a novel approach for recognizing Indian street names in the Bangla script, stands out with a remarkable 99.52% reliability. Introducing street name recognition in Indian languages addresses a significant gap, offering a valuable dataset for further research. Pal et al. [30] advance the recognition of Indian trilingual city names (English, Hindi, and Bangla) for postal automation, achieving an impressive overall recognition accuracy of 92.25%. This pioneering work lays the foundation for future trilingual city name recognition research. Pramanik et al. [31], a segmentation-free approach for recognizing handwritten Bangla city names, is presented, showcasing an accuracy of 98.86% with the ResNet50 CNN-TL architecture. This research represents a significant improvement over previous methodologies. Roy et al. [32] focus on recognizing handwritten multilingual, multiscript Indian city names, considering English, Bangla, and Devanagari. The research utilized CNN based techniques and achieved an impressive accuracy of 96%. This work highlights the need for advanced postal automation systems to handle multiple languages and scripts commonly used in India. Chatterjee et al. [33] address the challenge of recognizing handwritten city names in six major scripts for postal automation purposes. The bi-stage approach achieved an average script recognition accuracy of 99.07% and a city name recognition accuracy of 97.58%. The script-independent approach attained a recognition accuracy of 97.03% on a dataset comprising 807 classes, demonstrating promising results for postal automation applications. Thadchanamoorthy et al. [34] present a system for recognizing handwritten Tamil city names as a lexicon-driven word recognition problem. They achieved a reliability of 99.90% with error and rejection rates of 0.08% and 18.67%, respectively. This work significantly outperforms previous work on Tamil handwritten word recognition, improving substantially accuracy. Pramanik et al. [31] focus on developing a CNN-based model for recognizing handwritten district names in the "Gurmukhi" script. The proposed model achieves a remarkable accuracy, with a maximum validation accuracy of 99%. This research is precious for postal automation applications in India's multilingual and multiscript environment. The hybrid neuro-symbolic system, KBANN, was introduced by author Souici et al. [35] to recognize handwritten city names in Algerian postal addresses. The system combines perceptual features analysis and a hierarchical knowledge base, allowing faster training and improved accuracy.

In this extensive literature review, we have observed significant work and notable achievements in handwritten city name recognition, mainly focusing on various scripts and languages across multiple regions. However, there needs to be more research regarding Bangladeshi Bangla city name recognition, an area of particular importance given Bangladesh's unique linguistic and geographical landscape. In response to this gap, we embarked on developing a comprehensive dataset encompassing all districts in Bangladesh. We trained a recognition system tailored to the intricacies of Bangladeshi Bangla script. By undertaking this endeavour, we aim to fill this research void and provide valuable insights and resources for advancing handwritten city name recognition in Bangladesh. This initiative addresses the region's specific requirements. It contributes to the broader field of postal automation, offering a benchmark for future developments in Bangladeshi Bangla city name recognition and segmentation algorithms. Through this work, we aspire to support more efficient postal services, contributing to the overall progress of our region's infrastructure.

3 Dataset Description

The dataset presented in this study addresses the critical need for handwritten Bangladeshi Bangla city name recognition and encompasses a diverse collection of handwritten samples from all 64 districts of Bangladesh.

The dataset was meticulously curated to facilitate research and developments in city name recognition, mainly focusing on the unique characteristics of Bangladeshi Bangla script. The dataset was initiated by collecting handwritten samples from a demographically diverse 50 participants. Each participant was requested to provide three handwritten samples of city names corresponding to the 64 districts of Bangladesh. This meticulous effort resulted in an initial dataset comprising approximately 9600 handwritten city name samples.

This Fig. 3 shows an example of the data collection method used to gather handwritten city name samples for the dataset. To ensure uniformity and consistency in the dataset, all collected images were resized to a standard dimension of 400 pixels in width and 100 in height. This standardization ensures that all city name samples are of the same size and format, which is crucial for training and evaluating recognition systems. Data augmentation techniques were applied to the initial collection of city name samples to enhance the dataset's diversity and robustness.

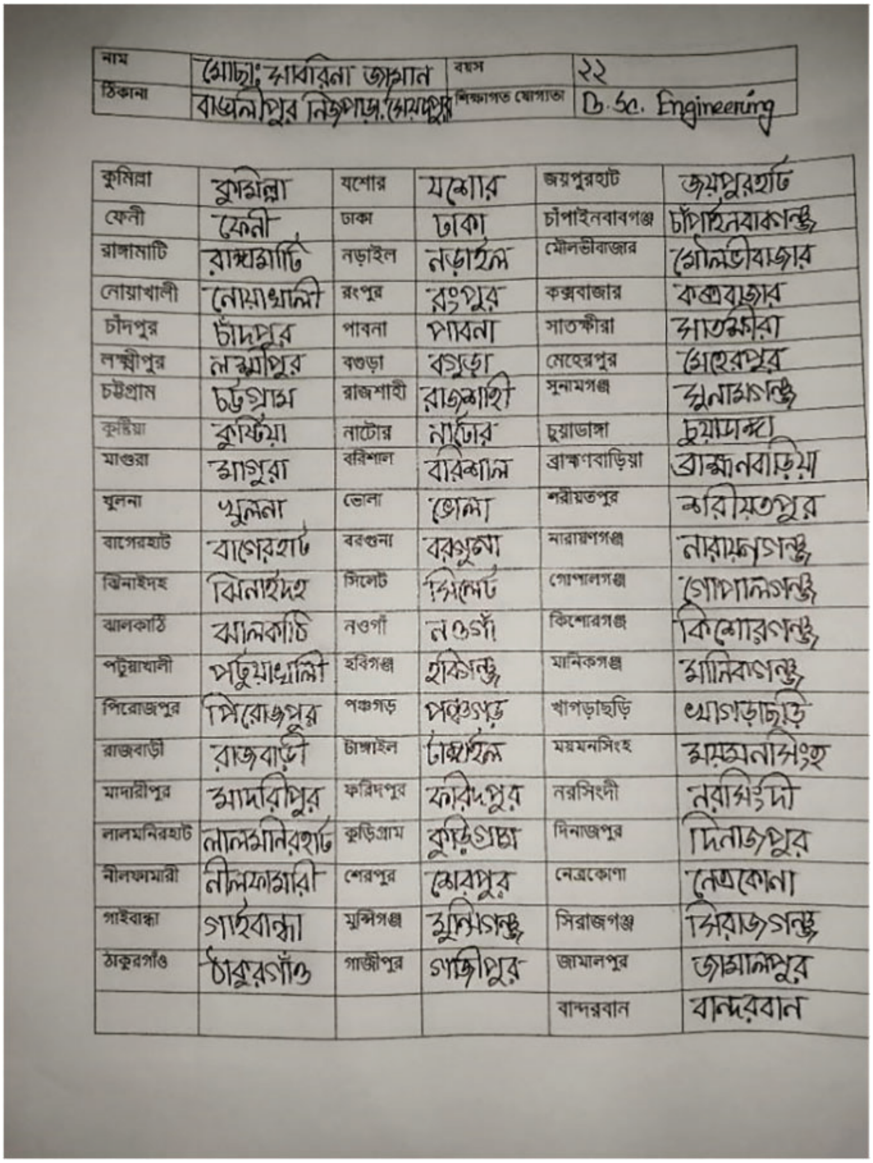
These augmentation methods included introducing variations such as image blurring, rotation, changes in brightness, and other transformations. As a result of this augmentation process, the dataset was expanded to contain a total of 32,000 images, with each of the 64 classes representing one of the districts containing 500 images. This augmentation increases the dataset's size and simulates the variability present in real-world handwritten city names. This Fig. 4 displays an example of an augmented image from one of the dataset's classes, illustrating the diversity and variations present in the dataset. This dataset represents a significant contribution to the handwritten city name recognition field, specifically tailored to the nuances of Bangladeshi Bangla script and the unique geographical regions of Bangladesh. It serves as a valuable resource for benchmarking recognition algorithms and promoting advancements in the automation of postal services in the region.

4 Methodology

In this study, we present a comprehensive methodology for recognizing handwritten Bangla city names and evaluating our proposed lightweight CNN model against several SOTA architectures, including VGG16, EfficientNetB0, ResNet50, InceptionV3, Xception, and DenseNet201. Our methodology encompasses data collection, preprocessing, model design, training, evaluation, and comparison. We emphasize the advantages of using our custom lightweight CNN over these well-established SOTA architectures. The benefits of using our lightweight custom CNN model over SOTA architectures are efficiency, specificity, customization, and interpretability. Our model is optimized for the targeted task, offering streamlined architecture, faster inference, and lower hardware requirements. It is trained explicitly on Bangla handwritten city names, resulting in higher recognition accuracy for this script. Moreover, its lightweight design provides adaptability, ease of customization, and improved interpretability. The reduced model size further conserves resources, making it a cost-effective choice for large-scale applications.

4.1 Basic Concepts of Convolutional Neural Network (CNN)

CNNs are a class of deep neural networks designed for processing grid-like data, such as images. They have revolutionized various computer vision tasks due to their ability to learn hierarchical representations of visual data automatically. Key components in CNNs include Conv2D, MaxPooling2D, GlobalAveragePooling2D, BatchNormalization, Dropout, and the final output layer.



নাম	মোছাঃ সাবরিনা জাহান	বয়স	২২
ঠিকানা	বাজেলাপুর নিজপাড়া.জয়পুরহাট	শিক্ষাগত যোগ্যতা	B.Sc. Engineering

কুমিল্লা	কুমিল্লা	যশোর	যশোর	জয়পুরহাট	জয়পুরহাট
ফেনী	ফেনী	ঢাকা	ঢাকা	চাঁপাইনবাবগঞ্জ	চাঁপাইনবাবগঞ্জ
রাঙ্গামাটি	রাঙ্গামাটি	নড়াইল	নড়াইল	মৌলভীবাজার	মৌলভীবাজার
নোয়াখালী	নোয়াখালী	রংপুর	রংপুর	কক্সবাজার	কক্সবাজার
চাঁদপুর	চাঁদপুর	পাবনা	পাবনা	সাতক্ষীরা	সাতক্ষীরা
লক্ষ্মীপুর	লক্ষ্মীপুর	বগুড়া	বগুড়া	মেহেরপুর	মেহেরপুর
চট্টগ্রাম	চট্টগ্রাম	রাজশাহী	রাজশাহী	সুনামগঞ্জ	সুনামগঞ্জ
কুষ্টিয়া	কুষ্টিয়া	নাটোর	নাটোর	চুয়াডাঙ্গা	চুয়াডাঙ্গা
মাগুরা	মাগুরা	বরিশাল	বারিশাল	ব্রাহ্মণবাড়িয়া	ব্রাহ্মণবাড়িয়া
খুলনা	খুলনা	ভোলা	ভোলা	শরীয়তপুর	শরিয়তপুর
বাগেরহাট	বাগেরহাট	বরগুনা	বরগুনা	নারায়ণগঞ্জ	নারায়ণগঞ্জ
ঝিনাইদহ	ঝিনাইদহ	সিলেট	সিলেট	গোপালগঞ্জ	গোপালগঞ্জ
ঝালকাঠি	ঝালকাঠি	নওগাঁ	নওগাঁ	কিশোরগঞ্জ	কিশোরগঞ্জ
পটুয়াখালী	পটুয়াখালী	হবিগঞ্জ	হবিগঞ্জ	মানিকগঞ্জ	মানিকগঞ্জ
পিরোজপুর	পিরোজপুর	পঞ্চগড়	পঞ্চগড়	খাগড়াছড়ি	খাগড়াছড়ি
রাজবাড়ী	রাজবাড়ী	টাঙ্গাইল	টাঙ্গাইল	ময়মনসিংহ	ময়মনাসিংহ
মাদারীপুর	মাদরিপুর	ফরিদপুর	ফরিদপুর	নরসিংদী	নরসিংদী
লালমনিরহাট	লালমানিরহাট	কুড়িগ্রাম	কুড়িগ্রাম	দিনাজপুর	দিনাজপুর
নীলফামারী	নীলফামারী	শেরপুর	শেরপুর	নেত্রকোণা	নেত্রকোনা
গাইবান্ধা	গাইবান্ধা	মুন্সিগঞ্জ	মুন্সিগঞ্জ	সিরাজগঞ্জ	সিরাজগঞ্জ
ঠাকুরগাঁও	ঠাকুরগাঁও	গাজীপুর	গাজীপুর	জামালপুর	জামালপুর
				বান্দরবান	বান্দরবান

Figure 3: An example of the data collection method used to gather handwritten city name samples for the dataset

Conv2D (Convolutional Layer): The Conv2D layer is fundamental in CNNs for extracting features from input data, particularly images [12]. It uses convolution operations to detect patterns in the data. A convolution operation involves sliding a small kernel or filter window over the input data, performing element-wise multiplications and summing the results. This produces feature maps that capture local patterns. The Conv2D operation is represented as Eq. (1):

$$O(i,j) = \sum_{m,n} I(i+m, j+n) \,.\, K(m,n) \tag{1}$$

where: $O(i, j)$ is the value at position (i, j) in the output feature map. $I(i + m, j + n)$ represents the input values at various locations. $K(m,n)$ denotes the learnable kernel weights. During training, CNNs adjust the kernel weights through backpropagation, effectively learning to recognize significant features.

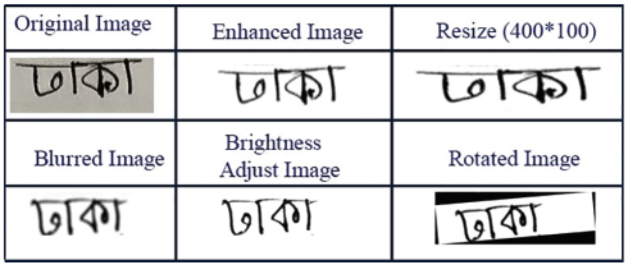


Figure 4: An example of an augmented image from one of the dataset's classes, illustrating the diversity and variations present in the dataset

MaxPooling2D (Pooling Layer): MaxPooling2D is a pooling layer that follows convolutional layers [12]. Its primary purpose is to reduce the spatial dimensions of feature maps while retaining essential information. Max pooling involves selecting the maximum value within a local region, thus reducing the size of the feature maps. The operation is described as Eq. (2):

$$O\,(i,j) = \max_{m,n} I\,(i + m, j + n) \tag{2}$$

MaxPooling2D helps create translation-invariant representations, making the network less sensitive to variations in the position of features within an image.

GlobalAveragePooling2D: GlobalAveragePooling2D is a layer used for spatial dimension reduction [36]. It computes the average of all values within each feature map. This results in a fixed-sized output, enabling compatibility with fully connected layers. It is particularly beneficial in image classification tasks where the network needs to produce a single prediction for the entire image. The operation of GlobalAveragePooling2D is represented as Eq. (3):

$$O = \frac{1}{N} \sum\nolimits_{i=1} I(i) \tag{3}$$

Here, O is the global average-pooled output, and N is the number of elements in the feature map.

BatchNormalization: BatchNormalization is a normalization technique applied within neural network layers [37]. It improves training stability and accelerates convergence by reducing internal covariate shifts. The normalization involves centering and scaling the activations for each mini-batch during training. The BatchNormalization transformation is defined as Eq. (4):

$$y = \frac{x - \mu}{\sqrt{\sigma^2 + \varepsilon}} \cdot \gamma + \beta \tag{4}$$

Here, x represents the input to the layer. μ is the mean over the mini-batch. σ is the standard deviation over the mini-batch. ε is a small constant added for numerical stability. γ and β are learnable scale and shift parameters. BatchNormalization helps prevent training issues, such as vanishing gradients and exploding activations.

Dropout: Dropout is a regularization technique used to mitigate overfitting [38]. Dropout randomly deactivates a fraction of neurons in a layer during training, effectively removing them from the network. This forces the network to learn more robust features and prevents it from relying too heavily on specific neurons. The dropout operation for a neuron's output is represented as Eq. (5):

$$O = \begin{cases} x \text{ with probability } p \\ 0 \text{ with probability } (1-p) \end{cases} \tag{5}$$

Here, x is the original output, and p is the dropout probability.

Outputs: The output layer in a CNN is responsible for producing the final predictions based on the features learned in previous layers. The activation function used in the output layer depends on the specific task. For multi-class classification, the softmax activation function is commonly applied to produce class probabilities [39]. The softmax function converts the logits (unnormalized scores for each class) into probabilities. It is defined as Eq. (6):

$$p\left(class_i\right) = \frac{e^{z_i}}{\sum_j e^{z_i}} \tag{6}$$

where $p\left(class_i\right)$ is the probability of class i, z_i is the logit for class i, and the denominator is the sum of logits for all classes.

4.2 Proposed Methodology

Fig. 5 illustrates the core structure of our Bangla handwritten city name recognition system, organized into three key components:

- We resize input images to 224 × 224 and divide them into training, validation and test sets.
- Augmentation techniques are exclusively applied to the training data, expanding the dataset's size while preserving semantic content.
- We introduce a lightweight CNN model optimized for feature extraction and classification. This model is assessed using the augmented dataset. Its performance will be benchmarked against SOTA models, including VGG16, EfficientNetB0, ResNet50, MobileNetV2, InceptionV3, Xception, and DenseNet201. This comparative evaluation will highlight the benefits of our lightweight custom model for Bangla handwritten city name recognition.

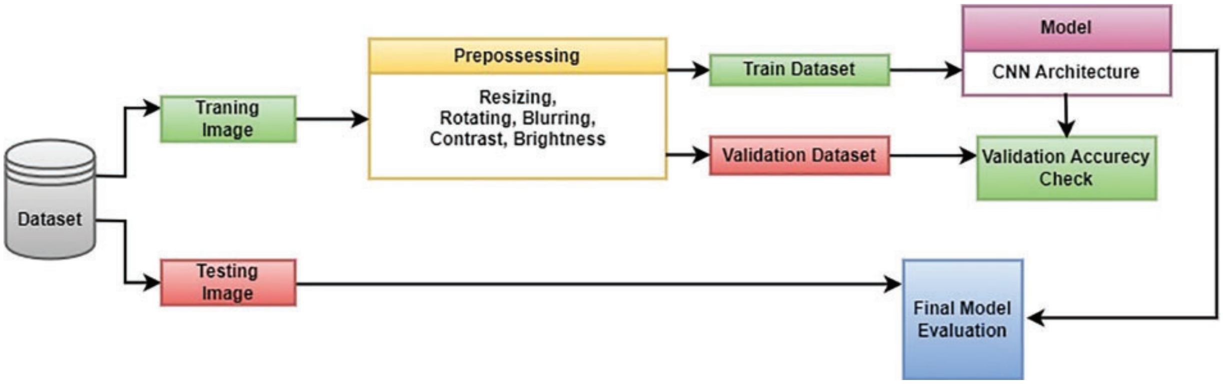


Figure 5: Proposed architecture of Bangla handwritten city name recognition system

The CNN architecture proposed for Bangla handwritten city name recognition has been meticulously tailored to address the unique demands of this task. This lightweight model is designed to extract relevant features and classify handwritten city names effectively. The architecture incorporates several crucial components to ensure its efficacy. Initially, the input images undergo preprocessing. They are resized to a standardized format of 224 × 224 pixels, ensuring all samples' image dimensions are consistent. This consistent formatting is vital for achieving optimal model performance.

Following the preprocessing, a rescaling layer is applied to the input data. This layer scales the pixel values within the range of [0, 1]. The rescaling operation is denoted as Eq. (7):

$$Rescaled\ Pixel\ Value = \frac{Original\ Pixel\ Value}{255} \tag{7}$$

This scaling is essential to ensure that the model can efficiently process the input data, as it helps standardize the pixel values, making them more amenable to neural network training. The CNN architecture starts with an initial convolutional layer. This layer employs 32 filters, each with a size of (3, 3). The filters have a stride of (2, 2), which means they move two pixels simultaneously when scanning the input image. The 'same' padding ensures that the output feature maps have the exact spatial dimensions as the input. The convolution operation with ReLU activation can be represented as Eq. (8):

$$Convolution(x, W) = ReLU\left(\sum (x * w) + b(x\ *\ W) + b\right) \tag{8}$$

Here, x represents the input feature map, W is the filter, b is the bias, and $\sum$ denotes the summation over the convolution operation.

BatchNormalization is incorporated after the convolution layer. This technique helps stabilize the training process by normalizing the activations within each mini-batch [37]. It has been shown to improve convergence and generalization in neural networks. The architecture also includes four convolution blocks. Convolution Block 1 involves two consecutive 2D convolutional layers, each equipped with 64 filters of size (3, 3) [40]. The 'same' padding and ReLU activation function are applied in both layers. Subsequently, a max-pooling layer is introduced to reduce the spatial dimensions of the feature maps. The pooling layer uses a pool size of (2, 2) and a stride of (2, 2). This down-sampling operation helps capture essential features and reduce the computational load. BatchNormalization is applied once more for stability, and a dropout rate of 25% is introduced to mitigate overfitting [38]. Convolution Block 2 shares the same structure as Block 1 but employs 128 filters. This filter increase allows the network to capture more complex and abstract features from the input images. Convolution Blocks 3 and 4 maintain the same architectural pattern, with 256 and 512 filters, respectively. These blocks play a crucial role in feature extraction, as they can capture high-level patterns and details from the handwritten city names. Global Average Pooling is applied at the end of the architecture. This layer aggregates spatial information across feature maps by calculating the average activation at each spatial location. This process results in a feature vector with a shape of (512) while retaining the number of channels. Global Average Pooling provides an efficient way to summarize the spatial information in the feature maps and is particularly useful for reducing model complexity. A densely connected layer follows Global Average Pooling with 512 units. The Rectified Linear Unit (ReLU) activation function is used in this layer. Additionally, a dropout rate of 50% is applied. These elements enhance model regularization, preventing it from fitting noise in the training data and improving its ability to generalize to unseen samples. The output layer consists of a dense layer with 64 units, corresponding to the number of output classes representing the different city districts in Bangladesh. The softmax activation function is employed in the output layer. It converts the raw model scores into class probabilities, facilitating multi-class classification. The output layer determines the predicted city name based on the input image. Throughout the architecture, batchNormalization and dropout functions are strategically placed to enhance the stability of the training process and promote generalization. BatchNormalization helps mitigate internal covariate shifts, which can hinder training. Dropout introduces a form of regularization, preventing overfitting by randomly deactivating a fraction of neurons during training. Fig. 6 and Table 1 shown in architecture details show the proposed CNN model.

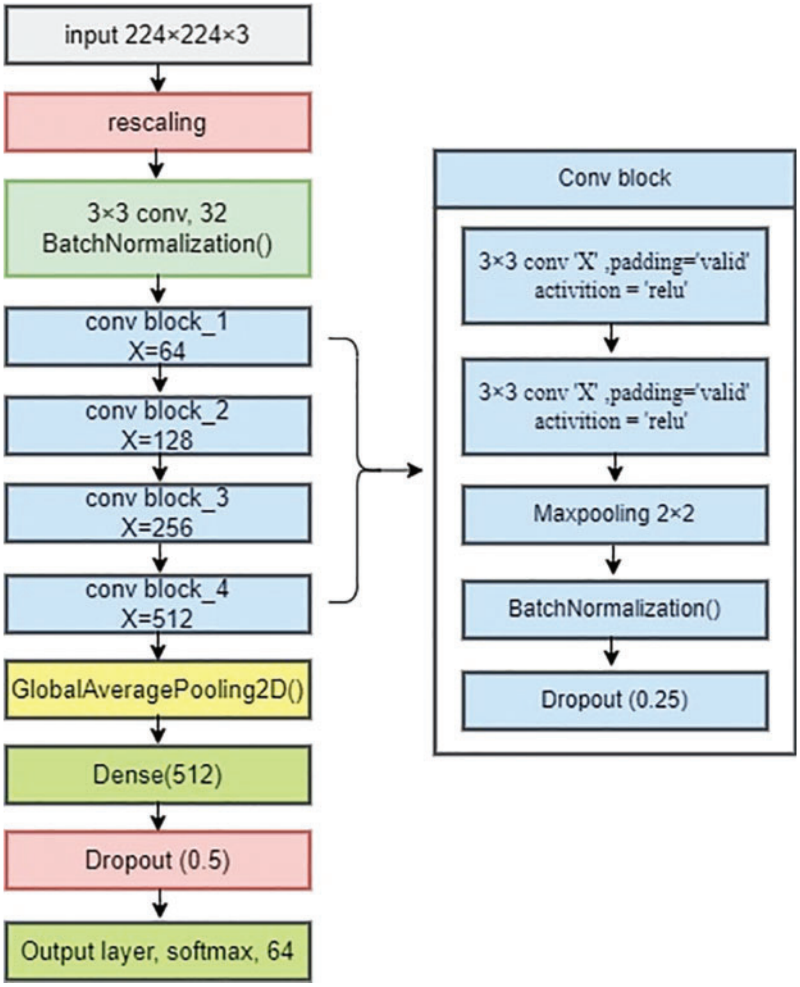


Figure 6: Proposed CNN model of Bangla handwritten city name recognition system

Table 1: Details and layer information of the proposed architecture

Layer (type)	Output shape	Param
input_1 (InputLayer)	[(None, 224, 224, 3)]	0
rescaling (Rescaling)	(None, 224, 224, 3)	0
conv2d (Conv2D)	(None, 112, 112, 32)	896
batch_normalization (Batch Normalization)	(None, 112, 112, 32)	128
conv2d_1 (Conv2D)	(None, 112, 112, 64)	18,496

(Continued)

Table 1 (continued)

Layer (type)	Output shape	Param
conv2d_2 (Conv2D)	(None, 112, 112, 64)	36,928
max_pooling2d (MaxPooling2D)	(None, 56, 56, 64)	0
batch_normalization_1 (BatchNormalization)	(None, 56, 56, 64)	256
dropout (Dropout)	(None, 56, 56, 64)	0
conv2d_3 (Conv2D)	(None, 56, 56, 128)	73,856
conv2d_4 (Conv2D)	(None, 56, 56, 128)	147,584
max_pooling2d_1 (MaxPooling2D)	(None, 28, 28, 128)	0
batch_normalization_2 (BatchNormalization)	(None, 28, 28, 128)	512
dropout_1 (Dropout)	(None, 28, 28, 128)	0
conv2d_5 (Conv2D)	(None, 28, 28, 256)	295,168
conv2d_6 (Conv2D)	(None, 28, 28, 256)	590,080
max_pooling2d_2 (MaxPooling2D)	(None, 14, 14, 256)	0
batch_normalization_3 (BatchNormalization)	(None, 14, 14, 256)	1024
dropout_2 (Dropout)	(None, 14, 14, 256)	0
conv2d_7 (Conv2D)	(None, 14, 14, 512)	1,180,160
conv2d_8 (Conv2D)	(None, 14, 14, 512)	2,359,808
max_pooling2d_3 (MaxPooling2D)	(None, 7, 7, 512)	0
batch_normalization_4 (BatchNormalization)	(None, 7, 7, 512)	2048
dropout_3 (Dropout)	(None, 7, 7, 512)	0
global_average_pooling2d (GlobalAveragePooling2D)	(None, 512)	0
dense (Dense)	(None, 512)	262,656
dropout_4 (Dropout)	(None, 512)	0
dense_1 (Dense)	(None, 64)	32,832

The compiled model is optimized using the Adam optimizer with a learning rate of 1e-3 [41]. The choice of optimizer and learning rate is crucial for efficient model training. The Adam optimizer employs the following Eqs. (9)–(13):

$$m_t = \beta_1 \,.\, m_{t-1} + (1 - \beta_1)\,.g_t \tag{9}$$

$$v_t = \beta_2 \,.\, v_{t-1} + (1 - \beta_2)\,.g_t^2 \tag{10}$$

$$m_t^* = \frac{m_t}{1 - \beta_1^t} \tag{11}$$

$$v_t^* = \frac{v_t}{1 - \beta_2^t} \tag{12}$$

$$\theta_t = \theta_{t-1} - \frac{\eta}{\sqrt{v_t^* + \varepsilon}} \,.\, m_t^* \tag{13}$$

The model uses sparse categorical cross-entropy as the loss function, as it is well-suited for multi-class classification problems. The sparse categorical cross-entropy loss is defined as Eq. (14):

$$L\left(y, \hat{y}\right) = -\sum\nolimits_{i=1}^{n} \sum\nolimits_{j=1}^{m} y_{ij} \,.\, \log(\widehat{y_{ij}}) \tag{14}$$

The training process spans 100 epochs, representing the number of times the model processes the entire training dataset. Early stopping is incorporated as a callback, which monitors the validation loss [42]. If the validation loss fails to decrease by a significant margin (in this case, a minimum delta of 1e-4) for a specified number of epochs (5 in this instance), the training process is halted. Early stopping prevents overfitting and ensures the model is trained effectively without unnecessary epochs.

4.3 The Architecture of the State-of-the-Art Model Used in Bangla Handwritten City Name Recognition

The architecture of the SOTA model used in Bangla handwritten city name Recognition is built upon a combination of renowned pre-trained models, including VGG16 [43], EfficientNetB0 [44], ResNet50 [45], InceptionV3 [40], Xception [46], and DenseNet201 [47]. This composite model is designed to leverage the strength of these architectures and achieve remarkable performance. In the initial stages, each of these pre-trained models, VGG16, EfficientNetB0, ResNet50, InceptionV3, Xception, and DenseNet201, is loaded with weights that were pre-trained on the ImageNet dataset [48].

This initial step aims to harness the rich knowledge these models have acquired from general image recognition tasks. The ImageNet weights provide a solid foundation for feature extraction. What sets this architecture apart is the subsequent fine-tuning process. The pre-trained weights of the base models are initially set as non-trainable by setting the trainable attribute to False. This ensures that the invaluable knowledge captured in the pre-trained weights is preserved during the initial phases of model training, allowing the model to function primarily as a feature extractor. After preserving the pre-trained weights, the model is ready for fine-tuning. Fine-tuning is essential in customizing the model for the specific task of Bangla handwritten city name recognition. During fine-tuning, the model adapts the pre-trained weights to this new recognition task by learning task-specific patterns from the dataset. The fine-tuning process optimizes the model's performance for city name recognition without completely overwriting the pre-trained knowledge. The model's architecture is summarized, including components such as VGG16, EfficientNetB0, ResNet50, InceptionV3, Xception, and DenseNet201. This summary provides an overview of the model's structure, allowing for a comprehensive understanding of its complexity. The output space of the model is tailored to match the requirements of the recognition task, with 64 output classes, each corresponding to a different city district in Bangladesh.

Data augmentation techniques are applied to the input images to prepare the model for fine-tuning. Data augmentation is a critical element that enhances the model's robustness and generalization capacity by introducing random transformations to the training data. Next, the input pixel values are rescaled to fall within the range of $(-1, +1)$. This rescaling is essential for specific pre-trained such as Xception, initially trained with inputs within this range.

Subsequently, the images are processed through the ensemble of pre-trained models. The base_model, which is an amalgamation of VGG16, EfficientNetB0, ResNet50, InceptionV3, Xception, and DenseNet201, is employed as a feature extractor, with a focus on fine-tuning. The training attribute is set to False to ensure that the base model operates in inference mode. The output from the base_model undergoes Global Average Pooling 2D. This layer calculates the average value of each feature map in the last convolutional layer, producing a feature vector with a shape of (512). This pooling operation effectively summarizes spatial information while maintaining an efficient model structure. Following Global Average Pooling, a dropout models, layer with a dropout rate 20% is added. Dropout is a regularization technique that randomly deactivates a fraction of neurons during training, thereby preventing overfitting and improving model generalization [30]. After the dropout layer, a dense layer with 512 units and ReLU activation is included [20]. This dense layer contributes to further feature transformation and is enriched with the Rectified Linear Unit (ReLU) activation function, defined as Eq. (15):

$$\int (x) = \max(0, x) \tag{15}$$

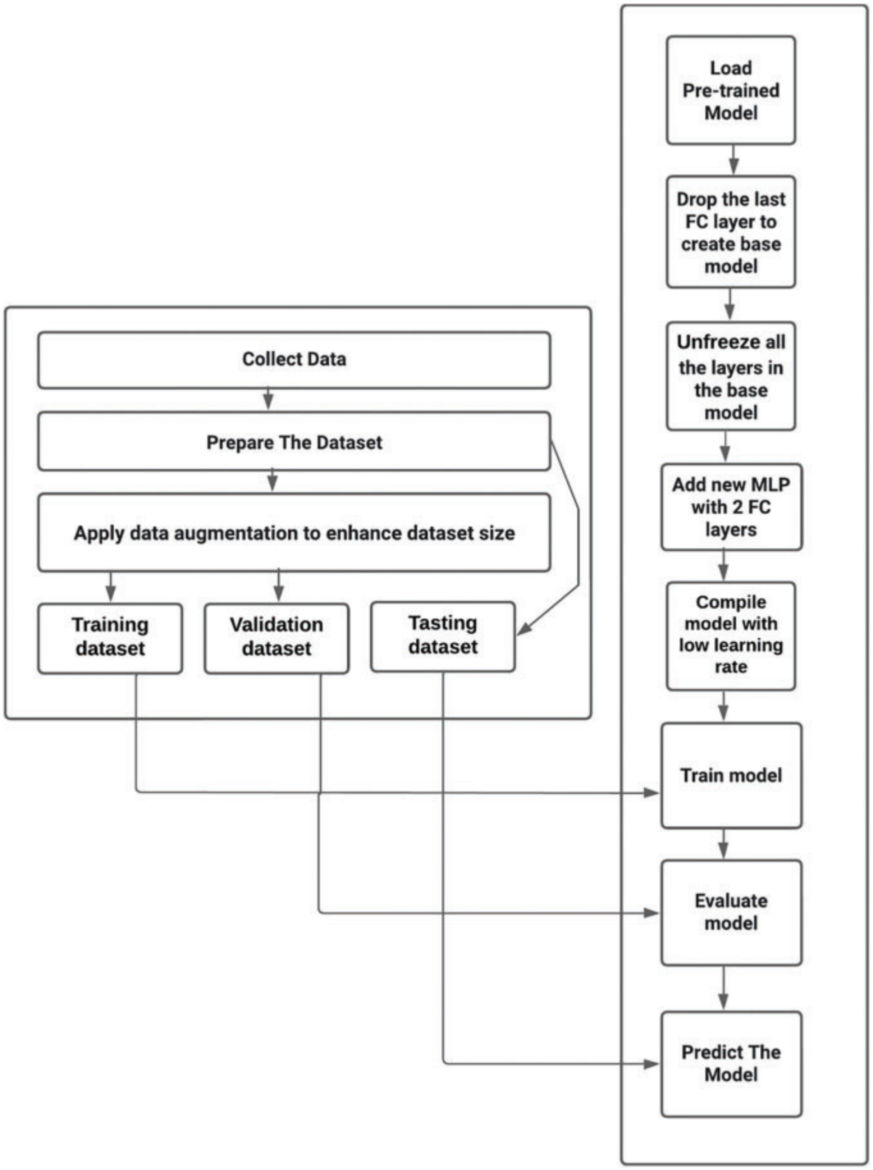

Collect Data
Prepare The Dataset
Apply data augmentation to enhance dataset size
Training dataset
Validation dataset
Tasting dataset
Load Pre-trained Model
Drop the last FC layer to create base model
Unfreeze all the layers in the base model
Add new MLP with 2 FC layers
Compile model with low learning rate
Train model
Evaluate model
Predict The Model

Figure 7: Proposed SOTA architecture of Bangla handwritten city name recognition system

The output layer is dense to complete the model, with 64 units aligned with the 64 classes representing different city districts. The softmax activation function converts the model's raw scores into class probabilities, a fundamental requirement for multi-class classification. The Architecture state-of-the-art (SOTA) is shown in Fig. 7. In fine-tuning, the model fine-tunes the base models to enhance their capability to recognize city names accurately. The fine-tuning process is supervised by the specific task requirements and the dataset, enabling the model to adapt to Bangla handwritten city name recognition nuances while retaining the knowledge acquired from ImageNet.

The compiled model is optimized using the Adam optimizer with a learning rate of 1e-3 [33]. The choice of optimizer and learning rate is crucial for efficient model training. The model uses sparse categorical cross-entropy as the loss function, as it is well-suited for multi-class classification problems. The training process spans 100 epochs, representing the number of times the model processes the entire training dataset. Early stopping is incorporated as a callback, which monitors the validation loss. If the validation loss fails to decrease by a significant margin (in this case, a minimum delta of 1e-4) for a specified number of epochs (5 in this instance), the training process is halted. Early stopping prevents overfitting and ensures the model is trained effectively without unnecessary epochs. This advanced model architecture represents the convergence of different SOTA models, fine-tuned for Bangla handwritten city name recognition. The fine-tuning process is pivotal, allowing the model to adapt to the specific task while capitalizing on the knowledge gleaned from general image recognition, culminating in a powerful and tailored solution for city name recognition.

The methodology outlined above lays the foundation for our proposed lightweight CNN model and the fine-tuning process, integrating the strengths of various SOTA architectures. Now, we transition to the results and analysis section, where we evaluate the model's performance in Bangla handwritten city name recognition and discuss the significance of our findings.

5 Result Analysis

The dataset used in this study plays a pivotal role in addressing the critical need for handwritten Bangladeshi Bangla city name recognition. It is a comprehensive collection of handwritten samples covering all 64 districts of Bangladesh, curated meticulously to serve as a vital resource for research and development in the field of city name recognition, with a particular focus on the unique characteristics of the Bangladeshi Bangla script.

When we turn our attention to the results, it is evident that established SOTA models and a custom CNN architecture have achieved remarkable performance in the challenging task of Bangla handwritten city name recognition. EfficientNetB0, VGG16, ResNet50, DenseNet201, InceptionV3, and Xception each demonstrated a high degree of accuracy and precision, making them promising candidates for practical applications in recognition technology. However, our custom CNN model, referred to as "Our CNN," emerged as the standout performer, showcasing near-perfect accuracy, precision, and recall. These results reinforce the effectiveness of the dataset, the significance of meticulous dataset engineering, and the potential of custom CNN architectures in real-world recognition tasks. With a test accuracy of 99.97% and an F1 score of 99.95%, it represents a robust solution for automating city name recognition. It holds considerable promise for broader applications beyond the scope of this study. These results underscore the dataset's role in advancing recognition technology and its far-reaching real-world implications. The Confusion Matrix is shown in Fig. 8. The Training and Validation Accuracy curve and Training and Validation Loss curve are shown in Fig. 9.

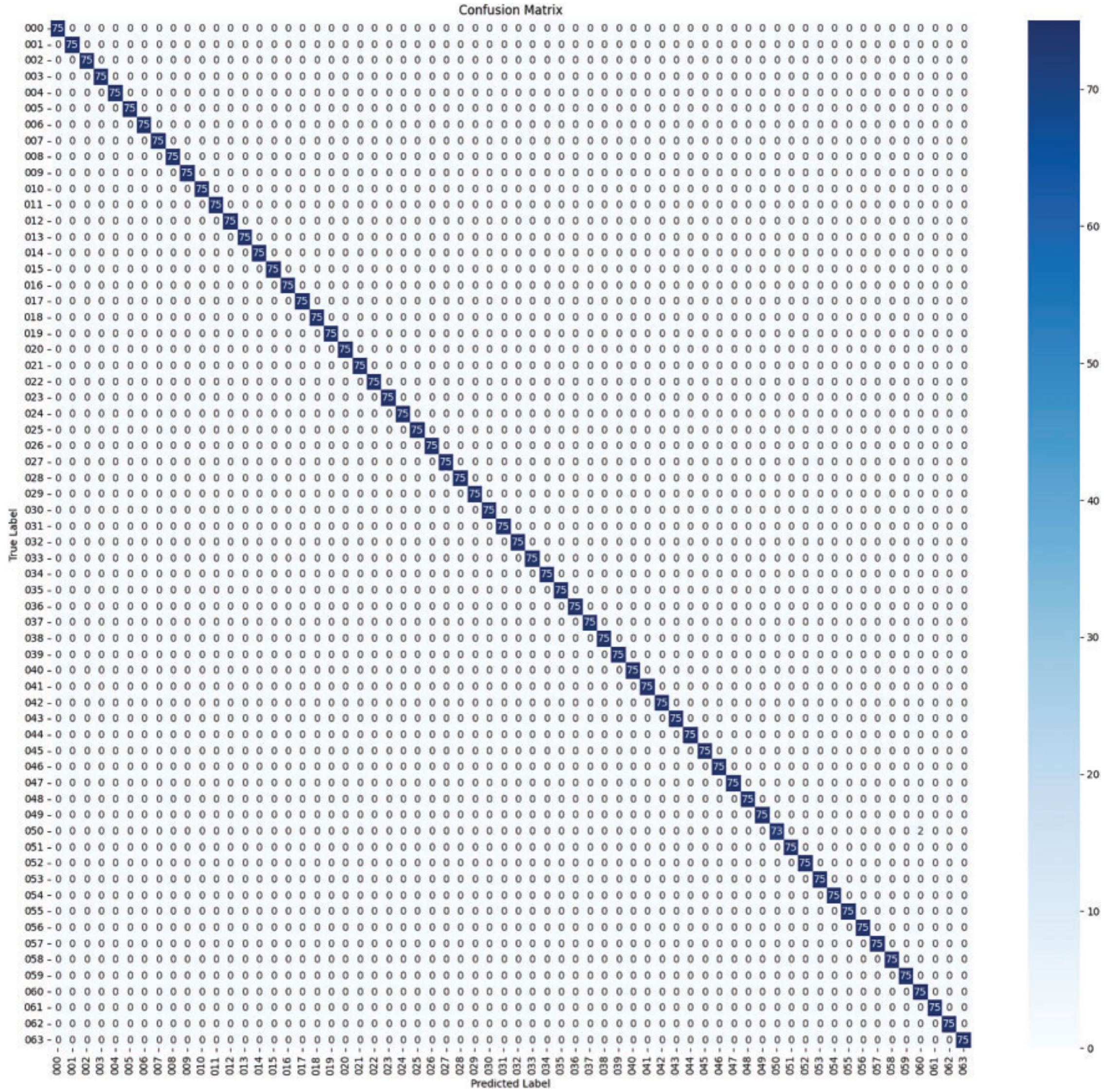
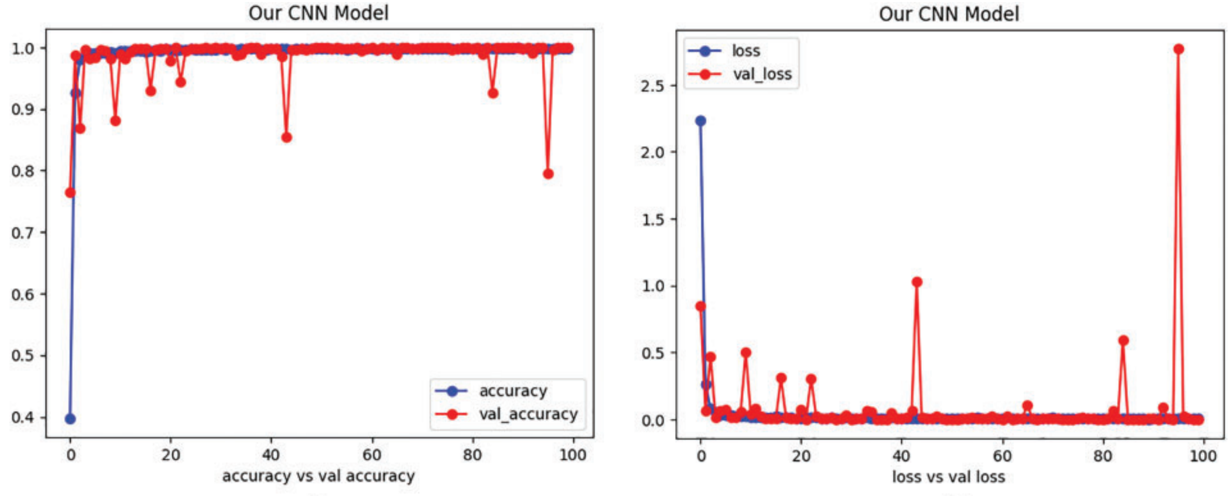


Figure 8: The confusion matrix for test dataset

In the analysis of the results, it is evident that the models, including SOTA architectures like EfficientNetB0, VGG16, ResNet50, DenseNet201, InceptionV3, and Xception, have displayed remarkable performance in the challenging task of Bangla handwritten city name recognition. EfficientNetB0 achieved impressive results with a high test accuracy of 99.59%. It maintained a strong balance between precision and recall, as indicated by an F1 score of 99.59%. The ROC-AUC score of 99.99% demonstrates its ability to distinguish between handwritten city names effectively. VGG16 also excelled with a test accuracy of 99.87%, complemented by strong F1, precision, and recall scores of 99.87%. Its ROC-AUC score, likewise approaching 99.99%, showcases its robust classification capabilities. ResNet50, while slightly lower in training accuracy, showed a test accuracy of 99.87% and an F1 score of 99.34%, making it a solid performer in city name recognition. Its precision and recall scores align closely with F1, indicating a well-balanced model. DenseNet201 achieved a high test accuracy of 99.75% and an impressive F1 score of 99.74%. It is precision and recall scores remained consistent with F1, underscoring its capacity to handle the recognition task effectively. InceptionV3 displayed an exceptional training accuracy of 99.91% and maintained strong performance on the validation and test sets. With an F1 score of 99.62%, it demonstrates reliable recognition capabilities. Xception, while slightly lower in validation and test accuracy, remained competitive with a test accuracy of 98.44%. Its F1 score, precision, and recall values were relatively balanced, ensuring robust recognition.



(a) (b)

Figure 9: (a) The training and validation accuracy curve and (b) Training and validation loss curve

In a comparative analysis against SOTA models, "Our CNN" stands out and excels in the domain of Bangla handwritten city name recognition. It achieves a remarkably high test accuracy of 99.97%, showcasing its proficiency in accurately identifying city names from handwritten samples. In contrast, the SOTA models, including EfficientNetB0, VGG16, ResNet50, DenseNet201, InceptionV3, and Xception, display test accuracies ranging from 98.44% to 99.75%. This significant gap in test accuracy underscores the superior recognition capabilities of "Our CNN." The F1 score, a pivotal metric indicating the harmonious balance between precision and recall, provides further evidence of the exceptional performance of "Our CNN." With an outstanding F1 score of 99.95%, "Our CNN" surpasses all the SOTA models, exhibiting F1 scores varying between 99.34% and 99.62%. The comparison is shown in Fig. 10 and Table 2.

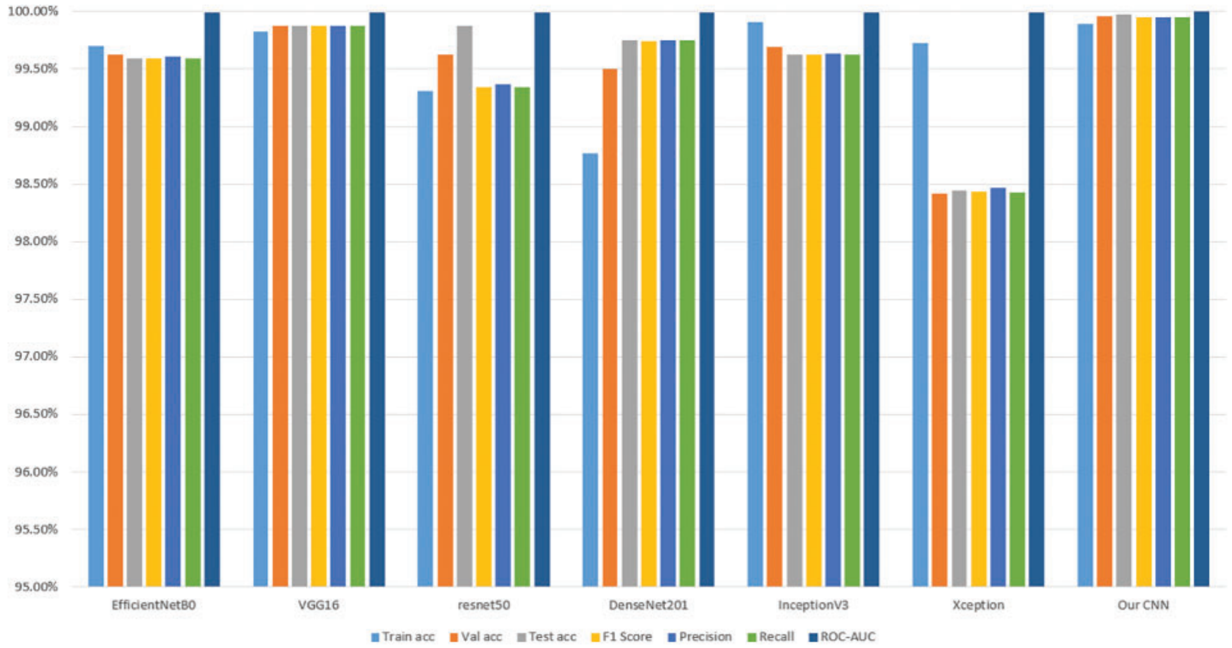


Figure 10: Comparison among SOTA and our model

Table 2: Comparison among SOTA and our model

Model	Train acc	Val acc	Test acc	F1 score	Precision	Recall	ROC-AUC
EfficientNetB0	0.997	0.9962	0.9959	0.9959	0.9961	0.9959	0.9999
VGG16	0.9982	0.9987	0.9987	0.9987	0.9987	0.9987	0.9999
ResNet50	0.9931	0.9962	0.9987	0.9934	0.9937	0.9934	0.9999
DenseNet201	0.9877	0.995	0.9975	0.9974	0.9975	0.9975	0.9999
InceptionV3	0.9991	0.9969	0.9962	0.9962	0.9963	0.9962	0.9999
Xception	0.9972	0.9842	0.9844	0.98435	0.9847	0.9843	0.9999
Our CNN	0.9989	0.9996	0.9997	0.9995	0.9995	0.9995	0.9999

This underlines the model's ability to achieve high precision and recall in recognizing Bangla city names. "Our CNN" consistently outperforms its counterparts, making it a standout performer in city name recognition. It demonstrates remarkable accuracy and recognition capabilities with a near-perfect test accuracy, exceptional F1 score, and high precision and recall. This impressive performance hints at the practical applications of "Our CNN," particularly in automating postal services, where precise city name recognition is essential. Consequently, these results contribute to the advancement of Bangla handwritten recognition technology and hold significant real-world implications for improving postal services in Bangladesh and similar contexts.

6 Discussion

We present a Lightweight CNN model for the recognition of handwritten Bangla city names for postal automation or document analysis. A dataset with diversity has been collected with 9600 samples taken from all the country's 64 districts considering data augmentation. It has achieved an extremely high accuracy of 99.97% recognition rate by suggesting a lightweight CNN model, thus proving to be superior to the existing models in efficiency as well as specificity. Among the methodological innovations in our approach is the CNN for Bangladeshi District Name recognition implemented with a lightweight of only 9 convolutional layers. On the other hand, a whole row of popular pre-trained models, such as VGG16, ResNet50, DenseNet201, EfficientNetB0, InceptionV3, and Xception, has a complex architecture with a considerable amount of layers. For example, about 215 is the average number of layers of EfficientNetB0, while VGG16 combines about 16 layers of convolution, ResNet50 uses 50 layers, DenseNet201 juxtaposes 201 layers, and InceptionV3 has been composed of 48 layers. This comparison underlines the simplicity and efficiency of our model that achieves competitive performance in recognition of Bangla handwritten city names. Developing recognition models and establishing benchmark datasets are pivotal in propelling research in this field. In its exploration of CNN-based handwritten city name recognition, this paper contributes to a broader narrative of automation and efficiency enhancements across numerous industries. In the forthcoming sections, we will embark on a comprehensive journey, delving into this innovative approach's methodologies, results, and implications. The tapestry of handwritten city name recognition, woven with the threads of CNN-based deep learning, holds immense promise in simplifying complex document-related processes and enhancing efficiency in our increasingly digitized world [18]. Table 3 provides a comparative analysis of test accuracy, model parameters, and model size for various deep learning models and our CNN model.

Table 3: Analysis of test accuracy, model parameters, and model size

Model	Test accuracy	Params	Size (MB)
EfficientNetB0	0.9959	5.3M	29
VGG16	0.9987	138.4M	528
ResNet50	0.9987	25.6M	98
DenseNet201	0.9975	20.2M	80
InceptionV3	0.9962	23.9M	92
Xception	0.9844	22.9M	88
Our CNN	0.9997	5.002M	19

7 Conclusion and Future Work

This study has presented a robust solution for Bangla handwritten city name recognition that is tailored to the distinct characters and forms of both the Bangladeshi Bangla script and the districts. It has undergone a meticulous process of curation and standardization, transforming it into a valuable asset available for advancing recognition technology. When measured against test performance, all results were over 99%, unlike the test accuracies of SOTA models: EfficientNetB0, VGG16, ResNet50, DenseNet201, InceptionV3, and Xception—as reported in previous research. However, the major highlight in the current work is our custom CNN model "Our CNN," demonstrating both accuracy and F1 score and highlighting its close-to-perfection recognition capabilities. Such findings highlight the vital necessity of dataset quality and point the way toward original CNN architectures in recognition tasks. The excellent performance of "Our CNN" also lays down ways of application in practice, especially with automation of postal service activities having a big focus on the proper naming of cities. Handwritten recognition technology has applications like regional administration, census data collection, and various postal delivery services. The study is a success as it opens a new field for future research in the domain of handwritten recognition technology, with elaborated algorithms, model architectures, and curation techniques for datasets being developed. Moreover, the results show potential to recognize handwritten text from other regional scripts and languages. Further research should look into ways to overcome challenges encountered in this research, including possible biases from limited handwriting styles, increasing generalizability to other scripts and regions, as well as exploring scalability and adaptability of the CNN architecture for broader recognition tasks. This way, it will be able to derive meaning from unevenly captured documents and thus work towards improved and broader applicability of recognition technology.

Acknowledgement: Finally, thanks to all those contributors of handwritten data who cooperated in all ways, because of that the collection becomes possible for the implementation of this research. Cooperation shows willingness and attitude towards all the people who got ready for the participation, also giving various types in handwriting to make the base. We extend every single thank and great cooperation to each one of them, since definitely this study would not have been possible without their help in the way we are receiving it.

Funding Statement: MMU Postdoctoral and Research Fellow (Account: MMUI/230023.02).

Author Contributions: Study conception and design: Md. Mahbubur Rahman Tusher, Md. Al-Hasan, Susmita Roy Rinky, Mehedi Hasan Jim; data collection: Md. Mahbubur Rahman Tusher, Md. Al-Hasan, Susmita Roy Rinky, Mehedi Hasan Jim; analysis and interpretation of results: Md. Mahbubur Rahman Tusher, Abu Saleh Musa Miah, Fahmid Al Farid, Sarina Mansor, Md. Abdur Rahim, Hezerul Abdul Karim; draft manuscript preparation: Md. Mahbubur Rahman Tusher, Abu Saleh Musa Miah. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: The data and code have been available to the following link https://github.com/tusher100/CIty-Name-Recognition (accessed on 21 March 2024).

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] M. Safiya, P. Kamakshi, T. M. Kumar, and T. Senthil Murugan, "Interpretation of handwritten documents using ML algorithms," in *Information and Communication Technology for Competitive Strategies (ICTCS 2021) ICT: Applications and Social Interfaces*. Jaipur, India: Springer, 2022, pp. 495–501.

[2] A. Vachon, L. Ordonez, and J. R. Fonseca Cacho, "Global postal automation," in *Intell. Syst. Appl.: Proc. 2021 Intell. Syst. Conf. (IntelliSys)*, Amsterdam, Netherlands, Springer, 2022, vol. 3, pp. 135–154.

[3] R. Mondal, S. Malakar, E. H. Barney Smith, and R. Sarkar, "Handwritten english word recognition using a deep learning based object detection architecture," *Multimed. Tools Appl.*, vol. 81, no. 1, pp. 975–1000, 2022. doi: 10.1007/s11042-021-11425-7.

[4] 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.

[5] M. R. Majumder, B. U. Mahmud, B. Jahan, and M. Alam, "Offline optical character recognition (OCR) method: An effective method for scanned documents," in *2019 22nd Int. Conf. Comput. Inf. Technol. (ICCIT)*, Dhaka, Bangladesh, IEEE, 2019, pp. 1–5.

[6] G. Kaur and T. Garg, "Machine learning for optical character recognition system," in *Machine Vision Inspection Systems*, New Jersey: U.S. Wiley Online Library, 2021, pp. 91–107.

[7] F. Abdurahman, E. Sisay, and K. A. Fante, "AHWR-Net: Offline handwritten amharic word recognition using convolutional recurrent neural network," *SN Appl. Sci.*, vol. 3, no. 8, pp. 1–11, 2021. doi: 10.1007/s42452-021-04742-x.

[8] D. Das, D. R. Nayak, R. Dash, B. Majhi, and Y. D. Zhang, "H-WordNet: A holistic convolutional neural network approach for handwritten word recognition," *IET Image Process.*, vol. 14, no. 9, pp. 1794–1805, 2020. doi: 10.1049/iet-ipr.2019.1398.

[9] T. T. Zin, S. Thant, M. Z. Pwint, and T. Ogino, "Handwritten character recognition on android for basic education using convolutional neural network," *Electronics*, vol. 10, no. 8, p. 904, 2021. doi: 10.3390/electronics10080904.

[10] M. A. Azad, H. S. Singha, and M. M. H. Nahid, "Zilla-64: A bangla handwritten word dataset of 64 districtsname of Bangladesh and recognition using holistic approach," in *2021 Int. Conf. Sci. Contemp. Technol. (ICSCT)*, Dhaka, Bangladesh, IEEE, 2021, pp. 1–6.

[11] N. Audebert, C. Herold, K. Slimani, and C. Vidal, "Multimodal deep networks for text and image-based document classification," in *Machine Learning and Knowledge Discovery in Databases: International Workshops of ECML PKDD 2019*, Würzburg, Germany: Springer, 2020, pp. 427–443.

[12] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Adv. Neural Inf. Process. Syst.*, vol. 25, 2012.

[13] T. Ghosh, S. Sen, S. M. Obaidullah, K. Santosh, K. Roy and U. Pal, "Advances in online handwritten recognition in the last decades," *Comput. Sci. Rev.*, vol. 46, no. 12, pp. 100515, 2022. doi: 10.1016/j.cosrev.2022.100515.

[14] P. Verma and G. Foomani, "Improvement in OCR Technologies in postal industry using CNN-RNN architecture: Literature review," *Int. J. Mach. Learn. Comput.*, vol. 12, no. 5, 2022.

[15] H. Kaur, M. Kumar, A. Gupta, M. Sachdeva, A. Mittal and K. Kumar, "Bagging: An ensemble approach for recognition of handwritten place names in gurumukhi script," *ACM Trans. Asian Low Resour. Lang. Inf. Process.*, vol. 22, no. 7, pp. 1–25, 2023. doi: 10.1145/3593024.

[16] N. M. Tahir, A. N. Ausat, U. I. Bature, K. A. Abubakar, and I. Gambo, "Off-line handwritten signature verification system: Artificial neural network approach," *Int. J. Intell. Syst. Appl.*, vol. 13, no. 1, pp. 45–57, 2021. doi: 10.5815/ijisa.2021.01.04.

[17] D. Nurseitov, K. Bostanbekov, M. Kanatov, A. Alimova, A. Abdallah and G. Abdimanap, "Classification of handwritten names of cities and handwritten text recognition using various deep learning models," arXiv preprint arXiv:2102.04816, 2021.

[18] T. Ghazal, "Convolutional neural network based intelligent handwritten document recognition," *Comput., Mater. Continua*, vol. 70, no. 3, pp. 4563–4581, 2022. doi: 10.32604/cmc.2022.021102.

[19] H. Kaur and M. Kumar, "Benchmark dataset: Offline handwritten Gurmukhi city names for postal automation," in *Document Analysis and Recognition*, Hyderabad, India: Springer, 2019, pp. 152–159.

[20] S. Mozaffari, H. El Abed, V. Märgner, K. Faez, and A. Amirshahi, "IfN/Farsi-Database: A database of farsi handwritten city names," in *Int. Conf. Front. Handwriting Recognit.*, 2008.

[21] S. Sharma, S. Gupta, N. Kumar, and H. Chugh, "Analysis of the proposed CNN model for the recognition of Gurmukhi handwritten city names of Punjab," in *Mobile Radio Commun. 5G Netw.: Proc. Second MRCN 2021*, Kurukshetra, India, Springer, 2022, pp. 267–279.

[22] M. A. Mahjoub, N. Ghanmy, and I. Miled, "Multiple models of Bayesian networks applied to offline recognition of Arabic handwritten city names," arXiv preprint arXiv:1301.4377, 2013.

[23] S. Sharma *et al.*, "Optimized CNN-based recognition of district names of Punjab state in Gurmukhi script," *J. Math.*, vol. 2022, no. 2, pp. 1–10, 2022. doi: 10.1155/2022/6580839.

[24] S. Barua, S. Malakar, S. Bhowmik, R. Sarkar, and M. Nasipuri, “Bangla handwritten city name recognition using gradient-based feature,” in *Proc. 5th Int. Conf. Front. Intell. Comput.: Theory Appl.*, Singapore, Springer, 2017, pp. 343–352.

[25] P. K. Prasad, P. Banerjee, S. Chanda, and U. Pal, “Bengali place name recognition-comparative analysis using different CNN architectures,” in *Comput. Vis. Image Process.: 5th Int. Conf.*, Prayagraj, India, Springer, 2021, pp. 341–353.

[26] M. Kumar, M. K. Jindal, R. K. Sharma, and S. R. Jindal, “Performance evaluation of classifiers for the recognition of offline handwritten Gurmukhi characters and numerals: A study,” *Artif. Intell. Rev.*, vol. 53, no. 3, pp. 2075–2097, 2020. doi: 10.1007/s10462-019-09727-2.

[27] M. Ghosh, S. Malakar, S. Bhowmik, R. Sarkar, and M. Nasipuri, “Memetic algorithm based feature selection for handwritten city name recognition,” in *Comput. Intell., Commun., Bus. Anal.: First Int. Conf., CICBA 2017*, Kolkata, India, Springer, 2017, pp. 599–613.

[28] S. Sahoo, S. K. Nandi, S. Barua, S. Malakar Pallavi, and R. Sarkar, “Handwritten Bangla city name recognition using shape-context feature,” in *Intell. Eng. Inform.: Proc. 6th Int. Conf. FICTA*, Singapore, Springer, 2018, pp. 451–460.

[29] U. Pal, R. K. Roy, and F. Kimura, “Handwritten street name recognition for Indian postal automation,” in *2011 Int. Conf. Doc. Anal. Recognit.*, Beijing, China, IEEE, 2011, pp. 483–487.

[30] U. Pal, R. K. Roy, and F. Kimura, “Multi-Lingual city name recognition for Indian postal automation,” in *2012 Int. Conf. Front. Handwriting Recognit.*, Bari, Italy, IEEE, 2012, pp. 169–173.

[31] R. Pramanik and S. Bag, “Handwritten Bangla city name word recognition using CNN-based transfer learning and FCN,” *Neural Comput. Appl.*, vol. 33, no. 15, pp. 9329–9341, 2021. doi: 10.1007/s00521-021-05693-5.

[32] R. K. Roy, H. Mukherjee, K. Roy, and U. Pal, “CNN based recognition of handwritten multilingual city names,” *Multimed. Tools Appl.*, vol. 81, no. 8, pp. 11501–11517, 2022. doi: 10.1007/s11042-022-12193-8.

[33] S. Chatterjee, H. Mukherjee, S. Sen, S. M. Obaidullah, and K. Roy, “City name recognition for Indian postal automation: Exploring script dependent and independent approach,” *Multimed. Tools Appl.*, vol. 83, no. 8, pp. 1–24, 2023. doi: 10.1007/s11042-023-16137-8.

[34] S. Thadchanamoorthy, N. D. Kodikara, H. Premaretne, U. Pal, and F. Kimura, “Tamil handwritten city name database development and recognition for postal automation,” in *2013 12th Int. Conf. Doc. Anal. Recognit.*, Washington, DC, USA, IEEE, 2013, pp. 793–797.

[35] L. Souici, N. Farah, T. Sari, and M. Sellami, “Rule based neural networks construction for handwritten Arabic city-names recognition,” in *Artif. Intell.: Methodol., Syst., Appl.: 11th Int. Conf.*, Varna, Bulgaria, Springer, 2004, pp. 331–340.

[36] M. Lin, Q. Chen, and S. Yan, “Network in network,” arXiv preprint arXiv:1312.4400, 2013.

[37] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” in *Int. Conf. Machine Learn.*, Lille, France, 2015, pp. 448–456.

[38] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: A simple way to prevent neural networks from overfitting,” *The J. Machine Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.

[39] C. M. Bishop and N. M. Nasrabadi, *Pattern Recognition and Machine Learning*. New York, NY, USA: Springer, 2006.

[40] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, “Rethinking the inception architecture for computer vision,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Las Vegas, NV, USA, 2016, pp. 2818–2826.

[41] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” arXiv preprint arXiv:1412.6980, 2014.

[42] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, “Improving neural networks by preventing co-adaptation of feature detectors,” arXiv preprint arXiv:1207.0580, 2012.

[43] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” arXiv preprint arXiv:1409.1556, 2014.

[44] M. Tan and Q. Le, “Efficientnet: Rethinking model scaling for convolutional neural networks,” in *Int. Conf. Machine Learn., PMLR*, Long Beach, CA, USA, 2019, pp. 6105–6114.

[45] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proc. IEEE Conf. Comput. Vis.Pattern Recognit.*, Las Vegas, NV, USA, 2016, pp. 770–778.

[46] F. Chollet, “Xception: Deep learning with depthwise separable convolutions,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Honolulu, HI, USA, 2017, pp. 1251–1258.

[47] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, “Densely connected convolutional networks,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Honolulu, HI, USA, 2017, pp. 4700–4708.

[48] O. Russakovsky *et al.*, “Imagenet large scale visual recognition challenge,” *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, 2015. doi: 10.1007/s11263-015-0816-y.