# A HYBRID DEEP LEARNING MODEL FOR HANDWRITTEN BANGLA AND ENGLISH DIGIT RECOGNITION

by

**Md. Ali Akbar**

**Student ID: 19METE004F**

A thesis submitted in partial fulfilment of the requirements for the degree of

MASTER of SCIENCE in ELECTRONICS AND TELECOMMUNICATION ENGINEERING

Department of Electronics and Telecommunication Engineering

CHITTAGONG UNIVERSITY OF ENGINEERING AND TECHNOLOGY

SEPTEMBER 2024

**Department of Electronics and Telecommunication Engineering**
**Chittagong University of Engineering and Technology**

The Thesis Title: "A Hybrid Deep Learning Model for Handwritten Bangla and English Digit Recognition" submitted by Md. Ali Akbar, Roll No. 19METE004F Session 2019-2020, has been accepted as satisfactory in partial fulfillment of the requirements for the degree of Master of Science in Electronics and Telecommunication Engineering on 05/09/2024.

**BOARD OF EXAMINERS**

1. ---------------------------------------------- Supervisor Chairman

Dr. Md. Saiful Islam
Associate Professor
Department of Electronics and Telecommunication Engineering
CUET.

2. ---------------------------------------------- (Ex-Officio) Member

Dr. Md. Azad Hossain
Professor & Head
Department of Electronics and Telecommunication Engineering
CUET.

3. ---------------------------------------------- Member

Dr. Md. Azad Hossain
Professor
Department of Electronics and Telecommunication Engineering
CUET.

4. ---------------------------------------------- Member

Dr. Md. Jahedul Islam
Professor
Department of Electronics and Telecommunication Engineering
CUET.

5. 05.09.২০২৪ ---------------------------------------------- External-Member

Dr. Quazi Delwar Hossain
Professor
Department of Electrical & Electronic Engineering
CUET.

# Declaration

I hereby declare that the work contained in this Thesis has not been previously submitted to meet the requirements for an award at this or any other higher education institution. To the best of my knowledge and belief, the Thesis contains no material previously published or written by another person except where due reference is cited. Furthermore, the Thesis complies with the PLAGIARISM and ACADEMIC INTEGRITY regulations of CUET.

Md. Ali Akbar

**Md. Ali Akbar**

19METE004F

Department of Electronics and Telecommunication Engineering (ETE)

Chittagong University of Engineering & Technology (CUET)

# Dedication

To my uncle **Junayed Faruk Babul**

&

To my teacher **Md. Mehedi Farhad**

# List of Publications

The following publications are a direct consequence of the research carried out during the elaboration of the thesis and give an idea of the progression that has been achieved.

**Conference:**

- Publication 1: M. A. Akbar and M. S. Islam, "Deep Convolutional Neural Network for Handwritten Bangla and English Digit Recognition," in *2021 International Conference on Electronics, Communications and Information Technology (ICECIT)*, Khulna, Bangladesh, 2021, pp. 1-4, doi: 10.1109/ICECIT54077.2021.9641127.
- Publication 2: M. A. Akbar and M. S. Islam, "Hybrid Deep Learning Model for Handwritten Bangla and English Digit Recognition," in *2023 International Conference on Information and Communication Technology for Sustainable Development (ICICT4SD)*, Dhaka, Bangladesh, 2023, pp. 129-133, doi:10.1109/ICICT4SD59951.2023.10303498.
- Publication 3: M. A. Akbar, M. S. Islam and M. A. A. Fardil "Deep CNN-Based Multi-Model Framework for Bangla Handwritten Digit Recognition." [Ready to submit]

**Journal Article:**

- Publication 4: M. A. Akbar, M. S. Islam, M. A. Samad, and K. Choi, "An Improved Hybrid Dilated CNN and K-Nearest Neighbors-based Approach for Handwritten Bangla Digit Recognition." [Under Review]
- Publication 5: M. A. Akbar, M. S. Islam, and M. A. A. Fardil, "A Modified Hybrid Dilated CNN Model based on Self Attention Layer for Handwritten English Digit Recognition." [Ready to submit]
- Publication 6: M. A. Akbar, M. S. Islam, and M. A. A. Fardil, "A Modified SqueezeNet Model for Image Classification." [Ready to submit]

# Approval by the Supervisor

This is to certify that MD. ALI AKBAR has carried out this research work under my supervision, and he has fulfilled the relevant Academic Ordinance of the Chittagong University of Engineering and Technology so that he is qualified to submit the following Thesis in the application for the degree of MASTER of SCIENCE in ELECTRONICS AND TELECOMMUNICATION ENGINEERING. Furthermore, the Thesis complies with the PLAGIARISM and ACADEMIC INTEGRITY regulations of CUET.

Saiful

-----------------------------------------------------

**Dr. Md. Saiful Islam**

Associate Professor

Department of Electronics and Telecommunication Engineering

Chittagong University of Engineering & Technology

# Acknowledgement

This research work reflects my efforts as well as my supervisor's initiative and continuous inspiration. However, before anything else, I would like to thank Almighty Allah for giving me the strength to complete this thesis.

I would like to express my profound thankfulness to my esteemed supervisor, Dr. Md. Saiful Islam, Associate Professor at the Department of Electronics and Telecommunication Engineering (ETE) at Chittagong University of Engineering and Technology (CUET), for his invaluable supervision, perceptive criticism, and encouragement. This undertaking is achievable because of his exceptional monitoring and unwavering cooperation. I am truly grateful that he allowed me to work with him.

I want to express my appreciation to Professor Dr. Md. Azad Hossain, Head of the Electronics and Telecommunication Engineering (ETE) Department at Chittagong University of Engineering and Technology (CUET), for all of his assistance along the way.

Additionally, I would like to thank the authorities of the Chittagong University of Engineering and Technology (CUET) as well as all of the staff members of the Department of Electronics and Telecommunication Engineering (ETE) for their support in providing me with administrative and technical assistance during my postgraduate studies.

Moreover, I would like to give thanks and love to my father Sultan Ahmad, mother Nargis Akter, and brother Md Ali Abbas Fardil for their continuous support and encouragement during this journey.

Lastly, I want to express my gratefulness to my parents for teaching me about the arts and sciences and for always encouraging me to pursue my passions.

**Author**

# Abstract

Convolutional Neural Network (CNN) is widely used for handwritten Bangla and English digit recognition. The problem with the conventional CNN is that it has an intricate and time-consuming convolution process to create a feature map and it requires a lot of computation time during the classification phase. This work primarily proposes substituting the convolution kernels of the classic CNN model with dilated convolution kernels to address the limitations in handwritten digit recognition. Although the computation time decreases, the Dilated CNN model's feature extraction part still performs inefficiently due to information loss, which lowers accuracy. Observing these above-mentioned problems, a Hybrid Dilated CNN (HDC) model is proposed by using dilated convolution kernels with different dilation rates to eliminate the detail loss problem of the Dilated CNN model. The proposed Hybrid Dilated CNN (HDC) model can achieve larger receptive fields and can extract distant features. Finally, the Support Vector Machine (SVM) and K-Nearest Neighbor (KNN) are applied as a classifier to increase the digit classification accuracy. The BanglaLekha-Isolated dataset of Bangla digits is used to verify the proposed methods and demonstrate that under the same conditions, the accuracy of the Dilated CNN, HDC, and the proposed HDC with KNN are 95.85%, 96.72%, and 96.80%, respectively. The proposed HDC model with KNN shows 100% accuracy in the case of the MNIST dataset of English digits. The HDC – KNN model's effectiveness in recognizing digits of both Bangla and English language is perfectly demonstrated by the values of Precision, Recall, and F1 Score parameters. More evidence of the HDC – KNN model's superiority comes from the values of TPR and FPR parameters that are calculated for each of the digit classes in the case of both languages. These experimental findings suggest that integrating the HDC with the KNN algorithm improves the effectiveness of recognizing handwritten Bangla and English digits.

# বিমূর্ত

হাতে লেখা বাংলা এবং ইংরেজি অঙ্ক শনাক্তের ক্ষেত্রে বহুল ব্যবহৃত একটি পদ্ধতি হচ্ছে Convolutional Neural Network (CNN)। প্রথাগত CNN এর ক্ষেত্রে পরিলক্ষিত সমস্যা হচ্ছে এটির feature map তৈরির প্রক্রিয়া অত্যন্ত জটিল ও সময় সাপেক্ষ এবং শ্রেণীবিন্যাস ধাপে এটির অধিক গণনার সময় প্রয়োজন হয়। হাতে লেখা অঙ্ক শনাক্তের ক্ষেত্রে সীমাবদ্ধতা সমূহ দূর করার জন্য এই গবেষণা প্রাথমিকভাবে প্রথাগত CNN এর convolution kernel সমূহ dilated convolution kernel দ্বারা প্রতিস্থাপিত করার প্রস্তাব করে। যদিও গণনার সময় হ্রাস পায়, তবে Dilated CNN মডেলটির শনাক্তকরণ চিহ্ন তৈরির ধাপটি তথ্য হারানোর কারণে অদক্ষতা প্রদর্শন করে, যা নিম্নগামী নির্ভুলতা প্রদর্শনের দিকে ধাবিত করে। উপরে উল্লিখিত সমস্যাগুলি পর্যবেক্ষণ করে, Dilated CNN মডেলের তথ্য হ্রাস সমস্যা সমাধান করার জন্য একটি বিভিন্ন dilation হার সমৃদ্ধ Hybrid Dilated CNN (HDC) মডেল প্রস্তাব করা হয়েছে। প্রস্তাবিত Hybrid Dilated CNN (HDC) মডেলটি বৃহত্তর গ্রহণযোগ্য ক্ষেত্রসমূহ অর্জন করতে পারে এবং ব্যবধানবিশিষ্ট শনাক্তকরণ চিহ্নসমূহ চিহ্নিত করতে পারে। অবশেষে, অঙ্ক শ্রেণীবিভাগ শনাক্তকরণ নির্ভুলতার হার বৃদ্ধির জন্য Support Vector Machine (SVM) এবং K-Nearest Neighbor (KNN) ব্যবহার করা হয়েছে শ্রেণিবিন্যাস-কারী হিসেবে। প্রস্তাবিত পদ্ধতি সমূহ পর্যালোচনা করার জন্য বাংলা অঙ্কের BanglaLekha-Isolated dataset ব্যবহৃত হয়েছে এবং একই পরীক্ষণ মানদণ্ডে, প্রস্তাবিত Dilated CNN, HDC ও KNN সংযুক্ত HDC মডেলসমূহ যথাক্রমে ৯৫.৮৫%, ৯৬.৭২% ও ৯৬.৮০% নির্ভুলতা প্রদর্শন করে। প্রস্তাবিত KNN সংযুক্ত HDC মডেলটি ইংরেজি অঙ্কের MNIST dataset এর ক্ষেত্রে ১০০% নির্ভুলতা প্রদর্শন করে। বাংলা এবং ইংরেজি উভয় ভাষার অঙ্কগুলি শনাক্তকরণের ক্ষেত্রে HDC – KNN মডেলটির কার্যকারিতা Precision, Recall ও F1 Score parameter সমূহের মান দ্বারা প্রদর্শিত হয়। উভয় ভাষার ক্ষেত্রে প্রতিটি অঙ্ক শ্রেণীর জন্য গণনাকৃত TPR ও FPR parameter এর মান সমূহ HDC – KNN মডেলটির আরও কার্যকারিতা প্রদর্শন করে। এই গবেষণালব্ধ ফলাফলসমূহ নির্দেশ করে যে HDC এবং KNN algorithm এর সংমিশ্রণ হাতে লেখা বাংলা এবং ইংরেজি অঙ্ক শনাক্তের ক্ষেত্রে অত্যন্ত কার্যকরী।

# Table of Contents

# List of Figures

# List of Tables

# Nomenclature

## Acronyms and Abbreviations

| | |
|---|---|
| AI | Artificial Intelligence |
| ANN | Artificial Neural Network |
| CNN | Convolutional Neural Network |
| DCNN | Dilated Convolutional Neural Network |
| HDC | Hybrid Dilated Convolutional Neural Network |
| DL | Deep Learning |
| ML | Machine Learning |
| MLP | Multi-Layer Perceptron |
| ReLU | Rectified Linear Unit |
| BN | Batch Normalization |
| MaxPool | Max Pooling |
| Conv | Convolution |
| SGDM | Stochastic Gradient Descent with Momentum |
| GAP | Global Average Pooling |
| FC | Fully Connected |
| RF | Receptive Field |
| SVM | Support Vector Machine |
| KNN | K-Nearest Neighbors |

# Chapter 1: INTRODUCTION

*Introduction, Motivation, Related Research, Problem Statement, Objectives, Contributions, and Thesis Outline are all included in this chapter. This chapter presents an overview of the several key components of this research.*

## 1.1 INTRODUCTION AND MOTIVATION

From ancient times to the recent zenith of the success of modern technology, human always depends on the written format of the language of their spoken words either for their purpose of communication or to keep records of communication or contract. Till today handwritten communication format is the safest and most trustworthy communication method for highly classified information. From era to era, region to region mankind depends on handwritten digits and characters for the very basic communication format. Except in more advanced countries almost all shopping malls, bazaars, and small grocery markets on earth deal with handwritten digits for calculation and also make vouchers containing the amount value in the form of digits. First-growing countries like Bangladesh are still highly dependent on handwritten vouchers, cash memos, etc. in small businesses all over the country. But, due to advances in technology and lifestyle in Bangladesh, there is also a trend to keep those vouchers, and cash memo recordings in digital format. So, a convenient way to do this is by using an optical character recognition system. Which reduces the time of typing again those documents. In those documents, the most sensitive information is money or currency value. Which are represented by handwritten Bangla or English digits as because almost all voucher and cash memos are handwritten. So, this work focuses on developing a system to recognize handwritten Bangla and English digits.

Besides, A timeless and inescapable part of human life is the use of handwritten language. The world will always have handwritten language, no matter how digital it grows. Despite the transition to a digital age, there are still some situations where paper and a pen must be used [1]. To cope with digital formats, there is always a necessity to convert them into digital versions. So, work in the field is ongoing for major languages to crack the problem. But, in the arena of the Bangla language, there is an absence of a state-of-the-art model to tackle this problem. This is also a main reason to focus this research on developing a highly efficient system to detect handwritten Bangla and English digits.

Moreover, there are various ways in which electronics and telecommunication engineering are strongly linked to handwritten digit recognition [2]-[9]. In electronics and telecommunications engineering, handwritten digit recognition is essential for several reasons:

- Efficiency and Automation: It makes it possible to automate tasks like digitizing handwritten papers, processing bank checks, and sorting postal mail. The time and effort needed for manual data entry and processing are greatly decreased as a result.
- Security and Authentication: Secure communication systems use handwritten digit recognition to authenticate handwritten passwords and to validate handwritten signatures. This improves the safety of conversations and transactions over the internet.
- Improved Human-Computer Interaction: It makes communication between people and machines easier. For example, handwritten input recognition in touch-screen devices enables more intuitive and natural user interfaces.
- Data Analysis and Interpretation: Handwritten data analysis in telecommunications can aid in the processing and interpretation of

information from a variety of sources, including handwritten notes or forms, which can be essential for making decisions based on data.

- Investigation and Creation: It acts as a testbed for deep learning and machine learning algorithms that are being developed and improved. Other fields of computer vision and pattern recognition also use systems like Convolutional Neural Networks (CNNs), which are used in handwritten digit recognition. These uses demonstrate how crucial handwritten digit identification is to improving user experience, security, and efficiency in the fields of electronics and telecommunications engineering.
- Developments in Machine Learning: The accurateness and proficiency of various recognition jobs can be improved by the expansion and improvement of machine learning algorithms for handwritten digit appreciation, which has wider applications in electronics. These applications highlight the importance of handwritten digit identification for enhancing security, efficiency, and user experience in the domains of electronics and telecommunications engineering.
- Embedded Systems: A variety of applications, including smart devices and Internet of Things gadgets, can use handwritten digit recognition algorithms implemented in embedded systems. For these systems to work properly, they frequently need to be able to recognize objects quickly and accurately.
- User Interface Design: Handwritten digit recognition enables more intuitive and user-friendly interfaces in electronic devices such as tablets, smartphones, and digital notepads. This makes natural input methods available, which improves the user experience overall.
- Security Systems: To authenticate handwritten passwords and verify signatures, secure electronic systems use handwritten digit recognition. This

strengthens the security of access control systems and electronic transactions.

- Data processing: For applications like automated form processing and check verification, electronics engineers create technology that can handle and analyze handwritten data.

Furthermore, handwritten digit recognition holds particular significance in the field of communications engineering due to several factors [2]-[9]:

- Computerized Data Entry: The process of entering handwritten data into digital systems can be computerized using automated data entry. Processing forms, questionnaires, and other paperwork that are frequently completed by hand benefits greatly from this.
- Effective Mail Sorting: Handwritten digit recognition is used by postal services to read and sort mail according to zip codes. This lowers the possibility of error and expedites the mail delivery process.
- Enhanced Security: It is used to authenticate handwritten passwords and to validate handwritten signatures in secure communication systems. This strengthens the security of electronic communications and transactions.
- Data analysis: The ability to read and analyze handwritten notes and forms using handwritten digit recognition is useful for data-driven decision-making in the telecommunications industry.

These applications demonstrate how handwritten digit recognition improves user experience, security, and productivity in the electronics and telecommunications industry. These references highlight the interdisciplinary aspect of handwritten digit recognition and emphasize how important it is to improving user experience, security, and efficiency in both electronics and telecommunications engineering [2]-[9]. This importance in electronics and telecommunication engineering is also a significant motivation for concentrating on developing an extremely effective system in this research to identify handwritten Bangla and English digits.

Overall, automated digit or character recognition technologies benefit users by providing a spontaneous technique for recognizing the digit and text from an image, handwritten page, or printed document and converting them into digital format for further use with ease method. Then the users can keep them for digital bookkeeping records, for document verification, or digital data analysis. The whole process can be denoted as image classification which is most fundamental and extensively used in computer vision [10]. But, in terms of handwritten digits; the whole process is pretty complex. Because handwritten differs from person to person, handwritten Bangla and English digits have properties of different types of curvature, different types of shapes, geometrical differences, positional anomaly, and similarity. So, there is a necessity to develop a perfect structure that can easily detect handwritten Bangla and English digits despite variation, writing style, and similarity or dissimilarity.

## 1.2 RELATED RESEARCH

A lot of work has also been done to date in digit and character recognition to fulfill this necessity in various languages. M. J. Hasan et al. [11] proposed a 14-layer deep CNN architecture which is designed and implemented as a feature extractor and SVM is utilized as a classifier to detect handwritten Bangla digits using a large number of training data using several datasets. The diverse data preprocessing technique is also implemented along with a different amalgamation of number of epochs and batch size. To distinguish and translate images of distinct Bangla handwritten characters into an editable electronic presentation, R. R. Chowdhury et al. [12] present a process of handwritten character recognition. M. R. Bashar et al. [13] projected a digit recognition structure grounded on histogram and windowing methods. The scanned image file of a handwritten numeral is used as input in this designed structure. Then, the image is pre-processed. The image's boundary region is identified, and an M x N vector is created to store all of the image's color values. As a result, a windowing technique is created to extract 16 x16 features from a M x N matrix and create a consistent feature. A histogram is generated from the feature and saved to a reference catalogue. The histogram of the recognized digit is attained in the same way for recognition, and a recognition apparatus compares it to the histograms of the reference database to decide whether or not the system identifies a digit. After using the windowing approach to extract consistent features from the scanned image, the features are applied to create a histogram. To identify Bangla handwritten characters, M. A. R. Alif et al. [14] suggest modifying the ResNet-18 construction (a convolutional neural network architecture). The suggested approach is used with discrete datasets of handwritten Bangla characters. A modified version of a Deep Convolutional Neural Network acknowledged as ResNet-50 is anticipated by S. Chatterjee et al. [15]. ResNet is chosen by the authors due to its substantial emphasis on dropout and batch normalization. Utilizing the BanglaLekha-Isolated dataset, a nine-layer DCNN model is suggested for categorizing Bangla numbers by S. M. Azizul Hakim and Asaduzzaman [16]. H. Zunair et al. [17] utilized a transfer learning strategy with a pre-trained VGG16 structure on the NumtaDB dataset using a larger number of epochs and modified parameters. O. Paul [18] shows an image pre-processing technique on the NumtaDB dataset for recognizing handwritten Bengali digits. Resize, cropping, grayscale conversion, etc. are implanted and an effective CNN model is developed. R. Basri et al. [19] projected a deep convolutional neural network constructed Bangla handwritten digit recognition scheme. This study examines how well several cutting-edge deep CNN approaches work when it comes to reading handwritten numbers. Four deep CNN architectures are taken into account, including the MobileNet, AlexNet, GoogleLeNet (Inception-V3), and CapsuleNet models. On a sizable, fair, and heavily modified standard dataset, these four deep CNNs have been tested. A.S.A. Rabby et al. [20] anticipated a model named Bornonet. In this study, a straightforward, lightweight CNN model for categorizing the 50 fundamental Bangla characters—11 vowels and 39 consonants—found in Bangla handwriting—is proposed. S. Jaman et al. [21] demonstrate how single digit, vowel modifiers, and compound characters can all be recognized using CNN. This study has looked at a CNN grounded handwritten Bangla character identification system. The recommended method for cataloguing and standardizing the handwritten character of photographs, as well as classifying other characters, uses CNN. Nearly 4,50,000 different handwritten typescripts in various styles are employed in this study. The suggested model exceeds some of the most popular techniques already in use and has been shown to have a high appreciation accurateness level. M. A. Azad et al. [22] propose an Autoencoder with a Deep CNN, that is denoted as DConvAENNet for identifying Bangla handwritten. For simple and clear implementation, S. Ahsan et al. [23] projected a method that is split into two portions. One portion of the derived dataset utilized in this work is used to expertise the system, and the other portion is utilized to examine it.

Despite digits and characters, several works have also been accomplished in Bangla word recognition. Using a feature signifier created by merging various tetragonal, cryptic, and perpendicular pixel mass histogram-based features, a comprehensive approach to handwritten word identification is established by S. Bhowmik et al. [24]. The multi-layer perceptron and support vector machine classifiers are used distinctly for each step of the appreciation process. The dissection of handwritten Bangla words into typescripts and also recognition of the equivalent might occasionally prove to be a very difficult subject for researchers due to its cursive nature. The segmentation work is made more difficult by the presence of a relatively big character set as well as modifiers, ascendants, descendants, and compound characters. So, a holistic method is implemented in this work, as it avoids this type of character-level segmentation. R. Pramanik and S. Bag [25] proposed a CNN-based transfer learning and FCN model for Bangla handwritten city name word appreciation. Initially, five distinct commercial classifiers are used in this effort to compare the performance of the various statistical feature sets that were extracted from word pictures. After that, five different CNN-TL architectures were used to examine how well they performed on holistic Bangla words: AlexNet, VGG-19, VGG-16, GoogleNet, and ResNet50. This last remark compared the outcomes of all the aforementioned experiments using a seven-layer FCN model. A. Hazra et al. [26] projected a CNN-based approach for recognizing the handwritten character of Bangla-Meitei Mayek. The initial design of this study changed, and it started from scratch to form a special CNN architecture, which has many recompenses over traditional machine learning methods and the exclusive capability to combine classification and feature extraction. Additionally, this article attempted to explain the mathematical basis for the deep learning (DL) model's use of non-linearity. The nonlinear activation layer, convolutional layer, fully connected layer, and pooling layer are the four layers that brand up the proposed CNN architecture. In this work, two freely manageable Bangla datasets—cMATERdb and ISI Bangla datasets—are utilized. The similar structure is validated on the "Mayek27" Manipuri Character dataset that has been anticipated. S. Bag et al. [27] proposed a system using structural decomposition for recognizing Bangla compound characters. This technique involves breaking down a composite character into its parts. The complex character is then identified by removing the primitives of convex shapes and applying a template-matching strategy. The inventiveness of this method lies in the design of suitable character disintegration rules for segmenting the character essential into stroke sections and arranging them for meaningful shape component extraction.

Several noteworthy works are also done in the segment of character recognition in different languages. N. Alrobah and S. Albahli [1] proposed an Arabic handwritten character appreciation system constructed on a hybrid deep learning archetypal. In this study, it is suggested that Arabic handwritten characters be recognized and developed by cleaning the cutting-edge Arabic dataset named Hijaa and building a hybrid Convolutional Neural Network (CNN) model employing classifiers from the eXtreme Gradient Boosting and Support Vector Machine categories. The Arabic character pictures are processed using CNN to extract their features before being sent to the machine-learning classifiers. It is possible to get a recognition rate of up to 96.3% for 29 classes, which is a significant improvement over the Hijaa dataset's already cutting-edge findings. F. Siddique et al. [28] proposed a handwritten digit appreciation system using a Convolutional Neural Network. The purpose of this work is to compare the classification accuracy of CNN across a series of concealed layer and epoch numbers and to observe the variance in classification accuracy. This experiment used the Modified National Institute of Standards and Technology dataset to evaluate how well CNN performed. Moreover, the backpropagation procedure and stochastic gradient descent are utilized to expertise the system. P. K. Singh et al. [29] introduce a handwritten digit recognition system focused on moment-based features. A handwritten digit identification method for detecting numbers inscribed in five common Indian writings—Indo-Arabic, Bangla, Roman, Devanagari, and Telugu—is shown in this study. For each digit sample, a 130-element feature customary has been computed that essentially combines six diverse categories of moments: moment invariant, geometric moment, affine moment invariant, zernike moment, legendre moment, and complex moment. Finally, the method is tested via a variety of classifiers on the CMATER and MNIST datasets, and after conducting arithmetical implication tests, it is found that the Multilayer Perceptron classifier performs better than the others. X. Niu and C. Y. Suen [30] proposed a handwritten digit recognition technique assembled on a novel hybrid CNN-SVM classifier. In this work, a hybrid model is created by combining the capabilities of two potent classifiers that have a history of successfully identifying different patterns: the Support Vector Machine and the Convolutional Neural Network. SVM functions as a recognizer and CNN as an instructible feature extractor in this structure. By automatically extracting information from raw pictures, the hybrid model makes predictions. Studies have been conducted with the popular MNIST digit database. Studies using the similar database have been compared, and this fusion has produced improved findings. Y. S. Chernyshova et al. [31] projected an outline for recognizing text lines in camera-captured images using two-step CNN. "On the device" manuscript line appreciation outline is presented in this study and is intended for embedded or mobile systems. This system views individual character recognition as a language-dependent problem and per-character segmentation as a problem independent of language. Thus, as an alternative of using image processing techniques for the dissection step or end-to-end ANN, the projected explanation is grounded on dual independent Artificial Neural Networks and dynamic program design. This work uses ANNs with a restricted set of instructible parameters to avoid overfitting and to encounter the strict memory size limits placed on implanted systems. This framework's main target is to distinguish low-quality photos of identification credentials with intricate backgrounds and a wide range of languages and typefaces.

N. Saqib et al. [32] suggest the two best models out of the twelve that were created, adjusting hyper-parameters to see which structures offer the finest accurateness for which dataset. The stages involved in developing an adaptive outline for OCR models are covered by P. H. Jain et al. [33]. Their goal is to investigate a workable way to tailor an OCR explanation for a particular dataset of English language arithmetical digits that they created for their study.

Additionally, they create a digit recognizer by training their convolutional neural network model on the MNIST dataset and comparing it to other models skilled on blends of the MNIST and custom digits. For CNN-based handwritten numeral appreciation, S. Ahlawat et al. [34] investigated a range of design parameters, including the quantity of stride size, layers, receptive field, padding, dilution, and kernel size. Furthermore, they assess diverse SGD optimization strategies to enhance the handwritten digit recognition presentation. Convolutional neural networks, which are presently the most advanced method for this problem, are used by A. Loquercio et al. [35] to develop a wide series of classification hypotheses, in addition to other conventional and cutting-edge machine learning techniques. They then employ Meta-Nets, a family of recently created and effective ensemble techniques, to combine them in the best possible way. All hypotheses are taken into consideration in the classifier parliament that results, regardless of their technique or correctness. Using non-linearities, such as local contrast standardization and rectification, is the single most significant component for decent correctness on object identification benchmarks, as demonstrated by K. Jarrett et al. [36]. They also demonstrate that two feature extraction stages produce more accurate results than one. Most astonishingly, they demonstrate that, with the right non-linearities and pooling layers, a two-stage structure with arbitrary filters can produce nearly a 63% identification frequency on Caltech-101. Ultimately, the results demonstrate that the system can achieve state-of-the-art presentation on the NORB dataset (5.6%) with supervised refinement, and that unsupervised pre-training tracked by supervised refinement yields decent exactness on Caltech-101 (> 65%) and the lowermost identified error rate on the unrefined, unembroidered MNIST dataset (0.53%).

A generalization of Drop Out, Drop Connect is presented by L. Wan et al. [37] as a way to standardize big fully connected layers in neural systems. A randomly chosen portion of activations inside each layer is set to zero when using Dropout for training. Instead, Drop Connect sets everything but an arbitrarily chosen subcategory of the network's weights to zero. As a result, a casual subcategory of the units in the layer before each unit provides input. According to T. G. Dietterich and G. Bakiri et al. [38], the error-correcting code methodology can yield accurate class probability estimations much like the other approaches. When combined, these findings show that output codes with error correction capabilities offer a versatile approach to enhance the efficiency of inductive learning systems on problems involving multiple classes. To solve the inadequacies of the CNN algorithm, A. A. Yahya et al. [39] make the following contributions: Firstly, the effective receptive field (ERF) size is computed after accounting for domain knowledge. They can choose a typical filter size and improve the CNN's classification accuracy by calculating the ERF's size. Second, having too much data produces inaccurate findings, which lowers the accuracy of the classification. Before beginning the data classification mission, data preparation is conducted to ensure the dataset is clear of any superfluous or inappropriate variable quantity to the goal variable. Thirdly, data augmentation has been suggested as a way to reduce training and validation mistakes while avoiding dataset limitations. Fourthly, they suggest incorporating improver white Gaussian noise with $\sigma = 0.5$ into the MNIST dataset to replicate the real-world ordinary factors that can have an impact on image quality. To recognize handwritten numbers from the MNIST dataset, S. Ahlawat, A. Choudhary, et al. [40] created a hybrid model that combines the strengths of Support Vector Machines, and Convolutional Neural Networks. The essential characteristics of both classifiers are combined in the suggested hybrid model. SVM functions as a binary classifier and CNN as an automatic feature extractor in their suggested hybrid model. The suggested model's approach is trained and tested on the handwritten digit dataset from MNIST. S. Kulkarni and B. Rajendran et al. [41] reveal supervised learning in Spiking

Neural Networks for the complexity of handwritten digit recognition using the spike-triggered Normalized Approximate Descent technique. Using four times smaller amount parameters than the state-of-the-art, their system demonstrates that using neurons functioning at sparse biological spike rates underneath 300Hz attains a classification correctness of 98.17% on the MNIST dataset. They offer several findings from in-depth numerical studies about how to optimize the network topology and learning parameters to increase accuracy. Additionally, they outline several methods for optimizing the SNN for memory- and energy-constrained computer hardware, such as reducing the precision of storing the synaptic weights and making approximations in the computation of neural dynamics. An effective structure with numerous convolutions, ReLU, and pooling layers was developed by A. Garg et al. [42]. It has a 98.45% accuracy rate when evaluated on the MNIST data set. Additionally, this model's accuracy is tested on a comparable random image data set, yielding noteworthy results.

N. Schaetti et al. [43] examine Echo State Networks' capacity to categorize MNIST database digits. A system is proposed by H. Zhao, H. Liu, et al. [44] that consists of CNN-based feature mining from the MINST dataset and arithmetical synthesis of several classifiers skilled on distinct feature sets. The feature sets are produced by applying feature assortment to the CNN-extracted novel feature set. The outcomes of the experiment demonstrate that the fusion of classifiers can attain a classification accuracy of above 98%. A unique bio-inspired spiking convolutional neural network skilled in an acquisitive, layer-wise manner is investigated by A. Tavanaei, A. S. Maida, et al. [45]. The convolutional layer, pooling layer, and feature sighting layer of the spiking CNN are both trained using bio-inspired learning techniques. To train the convolutional layer's kernels, a sparse, spiking auto-encoder that represents key photographic aspects is used. A probabilistic spike-timing-dependent plasticity

(STDP) knowledge instruction is utilized in the feature sighting layer. This layer uses WTA-threshold, leaky, integrate-and-fire (LIF) neurons to express complex photographic characteristics. Using clean and noisy pictures, the innovative structure is tested on the MNIST digit dataset. The convolutional layer is stack-admissible, according to intermediate results, meaning it can provision a multi-layer learning edifice. For clear images, the recognition presentation is higher than 98%. A procedure for handwritten digit recognition in the milieu of collaborative learning is put forth by H. Zhao, H. Liu, et al. [46] in an effort to promote diversity among various classifiers trained using various learning algorithms. Precisely, the Convolutional Neural Network architecture is utilized to extract visual properties of handwritten numbers. Additionally, individual classifiers that were trained using random forests and K nearest neighbors, respectively, are combined to form an ensemble one. S. Ali et al. [47] developed a convolutional neural network with rectified linear units (ReLU) activation for handwritten digit recognition using a Deeplearning4j (DL4J) outline. In an attempt to surpass ensemble designs in accuracy and save costs and operational complexity, S. Ahlawat et al. [48] suggest a CNN architecture. Additionally, they recommend an appropriate set of learning parameters for CNN design, which allows us to set a new absolute record for classifying MNIST handwritten digits.

In order to cut down on calculation time and storage space, S. Chakraborty et al. [49] suggest reducing the feature maps that are utilized to instruct the CNN. According to experimental results, the amount of feature maps used to train the CNN can be decreased without significantly compromising accuracy. A unique CNN model combined with the extreme learning machine (ELM) technique is created by S. Ali et al. [50]. By taking advantage of the rectified linear unit activation function in conjunction with the CNN unit as a feature extractor, this model prevails over the sigmoid activation function. As an image classifier, the ELM unit yields the exact symmetry needed to distinguish between handwritten digits. The MNIST database is utilized in the creation and training of the CNN-ELM model, which is built on the deeplearning4j (DL4J) framework. USPS test datasets and self-built handwritten digits were used to validate the model. Additionally, they added different hidden levels to the architecture and saw how the accuracy varied. The CNN-ELM-DL4J technique performs well than the traditional CNN models in standings of computational time and accurateness, according to the results. A unique fully homomorphic ELM (Homo-ELM) is proposed by W. Wang et al. [51] that allows cloud searching tasks to be performed in a fully protected environment without jeopardizing user confidentiality. They carry out a thorough experiment using the MNIST dataset in both local and cloud contexts to show how effective their method is. Through an empirical assessment of three typical handwritten character datasets—ISI-Kolkata Bangla numeral, MNIST, ISI-Kolkata Odia numeral, and a recently generated NIT-RKL Bangla numeral dataset—D. Das et al. [52] test the presentation of ELM with various settings. Ultimately, they extract a few optimal ELM configurations that might function as broad principles for creating ELM-based classifiers. Extreme Learning Machine is a revolutionary method that H. Malik and N. Roy et al. [53] focus on and test using the MNIST data set. The only motivation for their effort is to compare the two approaches once ANN and ELM are used to obtain results. Convolutional extreme learning machine with kernel (CKELM), a unique model for problem-solving that was based on DL, is proposed by S. Ding et al. [54]. DL takes too long during training, and KELM is not good at extracting features. To extract features and classify them, the CKELM structure adds alternative convolutional layers and subsampling layers to the hidden layer of the innovative KELM. The gradient algorithm is not used by the convolutional layer or subsampling layer to modify parameters because random weights have been found to produce decent results in some topologies.

Lastly, they conducted experiments using the MNIST and USPS databases. The FCM-CNN-H-ELM framework for data categorization, which integrates FCM-Based CNN and H-ELM, is presented by X. Xu et al. [55]. The experiments' outcomes demonstrate that, with a minimal number of training samples, the suggested training sample selection approach and classification framework can guarantee consistent, even higher, prediction results and drastically cut down on training time.

## 1.3 PROBLEM STATEMENT

Most character recognition research relies on manually extracting features, which are then fed into corresponding classifiers or done entirely with CNN. Manual feature extraction is tricky and costs enormous labor. CNN can solve the manual feature extraction problem & can also do classification. But, in the section on feature acquisition, recent studies show that CNN cannot be able to detect some kind of the features such as Spatial analysis, Moment-Based Features: Geometric Moments, Moment Invariants, Affine Moment Invariants, Zernike Moments, & Slope and Curvature Function, etc. which are relevant to the structure of handwritten digits [56]. Besides, CNN consumes too much computing resources due to its convolution operation technique and Multi-Layer Perceptron is grounded on the Empirical Risk Minimization technique, which attempts to lessen the errors in the training set. As a result, the back-propagation algorithm stops training as soon as it finds the first separating hyperplane. This scenario results in the CNN's accuracy loss [30].

## 1.4 OBJECTIVES

Observing and analyzing those previously done works mentioned in section 1.2 (Related Research), especially, their outcomes and drawbacks this work proposes several models and evaluates their drawbacks. Every problem identified in a model is solved in the next model. By continuing the process this work tried to cover every available point i.e., time consumption, computing power consumption, more feature extraction, and training accuracy in the field of handwritten Bangla and English digit recognition. The main objectives of this research are mentioned below-

a. To develop an effective pre-processing method based on color & brightness correction, geometric transformation (i.e., scaling, translation, etc.), filtering (i.e., smoothing, edge detection, sharpening, etc.) for the respective dataset images.

b. To develop an effective feature extraction technique mainly focused on different writing styles, fonts, curves, arrangements, etc.

c. To develop an efficient classification system where CNN, Dilated CNN, HDC, KNN, and SVM can be evaluated for classification with proper tuning of layers focusing on less computational time consumption and more accuracy.

d. To design and adjust the parameters and layers of every step of this work in an industrious way so that it can provide the highest accuracy for industrial and consumer applications.

## 1.5 CONTRIBUTIONS

This research work's major contributions are:

a. An effective pre-processing method based on color & brightness correction, geometric transformation (i.e., scaling, translation, etc.), and filtering (i.e., smoothing, edge detection, sharpening, etc.) is developed in this work for handwritten digit datasets.

b. A dilated convolution neural network is implemented in the field of handwritten Bangla digit classification by stacking the convolutional kernel of traditional CNN. The embedded dilated convolution layers enlarge the receptive field by inserting holes in the kernel.

c. An improved Hybrid Dilated CNN (HDC) model is introduced in this study to extract more features & eliminate the time consumption and accuracy reduction problem of CNN and Dilated CNN models respectively. Different dilation rates are applied in convolution layers to design the HDC model.

d. This work also utilizes SVM and KNN algorithms with the proposed HDC model to fully utilize the capabilities of machine learning technology. The proposed technique outclasses several state-of-the-art methods for classifying handwritten Bangla and English digits.

## 1.6 THESIS OUTLINE

The remaining different portions of this thesis are organized in different chapters in this book. The contents of these chapters are elaborately discussed as follows:

Chapter 2 describes the historical background of the number system and derivation of the present handwritten digits format of the Bangla and English languages. The fundamental tools of this research i.e. Neural Networks, Convolutional Neural Networks (CNN), Dilated CNN, and Hybrid Dilated CNN (HDC) algorithm's technical insights are expressed in this chapter. The theoretical background and basic principles of the utilized classifiers in this research are also mentioned.

Chapter 3 is based on methodology, research design, dataset, dataset pre-processing, and design & development of the proposed models. The whole research process, including the dataset used for justification and dataset upgrades, is covered in this chapter. This chapter also provides a brief discussion of the models developed in this research. This chapter provides a detailed expression of all five of the suggested models, including information on the design and development process and simulation results. This chapter also presents the first three models' mathematical parameter evaluations. The description of hybrid machine learning classification models brings this chapter to a close.

Chapter 4 analyses the findings of the suggested models. Additionally, this chapter demonstrates how these results compare to current existing models constructed utilizing the MNIST, BanglaLekha-Isloated, and other datasets.

Chapter 5 provides an overview and assessment of the works accomplished in this research, as well as an explanation of the scope of future research on this area of study. Additionally, this chapter discusses the implications and limitations of this research.

# Chapter 2: THEORETICAL BACKGROUND

*The historical background of the Number System and the derivation of the present handwritten digits format of the Bangla and English language are described in this chapter in section 2.1. The fundamental tools of this research i.e. Neural Networks, Convolutional Neural Networks (CNN), Dilated CNN, and Hybrid Dilated CNN (HDC) algorithms are expressed in this chapter with sufficient technical insights in section 2.2. The theoretical background and basic principles of the SoftMax, SVM, and KNN classifier utilized in this work are described in this chapter in section 2.3.*

## 2.1 NUMBER SYSTEM AND HANDWRITTEN DIGITS

Early civilizations devised several methods for writing down numbers as they advanced. Numerous systems, such as the Hebrew, Egyptian, and Greek numeric systems, were essentially just tally marks expanded upon. They represented bigger values with a variety of symbols. For instance, a water lily symbolized 1000 and a looped rope represented 100 in the Ancient Egyptian structure. 300 would have been represented as three looped ropes in the Ancient Egyptian system since each symbol was multiplied as many times as needed and added together. However, even with this mechanism in place, writing big numbers was still a laborious process [57].

One thing unites all early number systems. To record a single number, they need someone to write down a lot of symbols, and for every larger number, they need someone to produce additional symbols. Using a positional system, one can reuse the same symbols by giving them distinct values according to where they fall in the sequence. Positional notation was separately established by many civilizations, including the Chinese, the Aztecs, and the Babylonians. Indian mathematicians achieved perfection in the decimal (or base ten) positional system by the 7th century, allowing any number to be represented with just ten distinct symbols. It was introduced throughout Europe during the following several centuries by Arab conquerors, academics, and traders. The number 0 was a significant innovation of this specific system, which the Mayans also separately invented. The absence of a zero in earlier positional notation systems made it difficult to discern between, for example, 63 and 603 or 12 and 120. It is easier for everyone to understand and write down numbers when 0 is present and used [57].

In mathematics, the decimal system is a positional numeral system that utilizes 10 as its base and requires 10 distinct numerals, which are the numbers 0, 1, 2, 3, 4, 5, 6, 7, 8, 9. To depict decimal fractions, a dot, or decimal point, is also needed. The numerals used to represent numbers in this scheme have varied place values based on their location. Since the 12th century CE, when this number system and its corresponding mathematical procedures were introduced to the West, they have provided the foundation for the advancement of Western science and commerce [58].

Thus, the perfect idea of the digit is derived and used from era to era for numerous purposes in handwritten format and adopted in every language including Bangla and English.

## 2.2 NEURAL NETWORKS

Neural networks are structures, or machine learning algorithms, that mimic the ways in which biological neurons cooperate to identify patterns, assess options, and make judgments. This enables them to form opinions in a manner akin to that of the human brain. Any neural network is composed of layers of nodes, or artificial neurons, which include an input layer, one or more hidden layers, and an output layer. Each node is linked to other nodes and has a weight and threshold assigned to it. If a node's output surpasses a predetermined threshold value, it becomes active and transmits data to the network's next tier. If not, no data is passed to the next tier of the network. Neural networks must get training data in order to learn and improve in accuracy over time. Once data has been accurately rectified, they are useful apparatus in computer science and artificial intelligence that let us quickly categorize and cluster data. Whenever speech recognition and picture recognition tasks are contrasted with manual identification by human experts, the former can be finished in minutes, while the latter can take hours [59].

A lot of work has been accomplished to date to upgrade and make more efficient the classic Artificial Neural Network and a big success is developing Convolutional Neural Network. This research work tries to upgrade the present state-of-the-art Convolutional Neural Network for better performance in the field of handwritten Bangla and English digit recognition.

### 2.2.1 Convolutional Neural Networks (CNN)

For handwritten Bangla digit appreciation, primarily a CNN structure is developed in this work. The elementary structure of a CNN is the composition of an input layer, hidden layer, and output layer just like in a traditional artificial neural network. But the difference is with traditional networks is that CNN can directly take images as input and capable of obtaining the feature of the input image and finally can do the classification task [60]-[61]. A basic overview of a CNN network is given in Fig. 2.1 for better understanding.

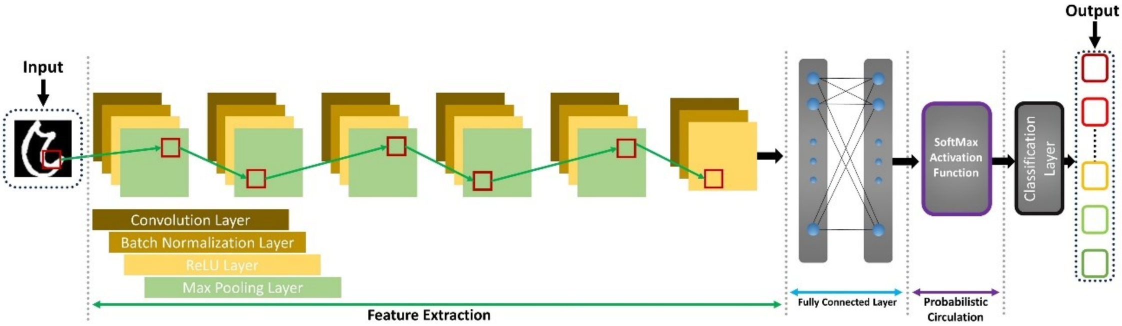


Fig. 2.1: Framework of proposed CNN model

From the respective of Fig. 2.1, it is observable that CNN is based on several mechanisms. The first mechanism is feature extraction which is typically based on a blend of the Convolutional Layer, Batch Normalization Layer, ReLU Layer, and Max pooling Layer or Average pooling Layer. Among these layers, the Convolutional Layer is more complex in operation [62]. It stores several filters or kernels and modifiable parameters to perform. It applies a filter on an image to generate a feature map that identifies the existence of detected features in the given image. A convolutional kernel links to a neuron. The size of the convolutional kernel is defined as the receptive field. The used filter is usually smaller than the input data and it generates values in singular form which is also known as scalar product. Then the filter is applied all over the participation image from leftward to right and upper to bottommost to create a feature map. This process can be expressed as:

$$R_w = \left(\frac{i_w - n + 2p}{k}\right) + 1 \qquad (2.1)$$

$$R_h = \left(\frac{i_h - n + 2p}{k}\right) + 1 \qquad (2.2)$$

Where the resulting feature graph size is the value of $R_w$ and $R_h$. The width and height of the input image size are represented by $i_w$ and $i_h$. Filter migration steps from leftward to right and top to bottommost are expressed by the value of $k$. The number of pixels covered by the steps is denoted by $p$.

Then batch normalization technique is applied in CNN between the layers. It applies by dividing the full data set into several mini-batches. The foremost resolution of this layer is to train the model in less time and make the model more stable through normalization of the input data from the preceding layer by performing re-centering along with re-scaling.

Again, to lessen the dimensions of the input image for successful training and to gain access to more feature information pooling layer is used. The two most frequently utilized pooling methods are max pooling and average pooling. If there is an image of 4 * 4 pixels and the applied pooling filter is 2 * 2. Then the filter starts from the top left corner covering the first four blocks of an image and reduces it to one block, then shifts to the next block. By repeating the whole process pooling creates a small feature map with a smaller RF [63]-[64]. The whole calculation process for both types is shown in Fig. 2.2.

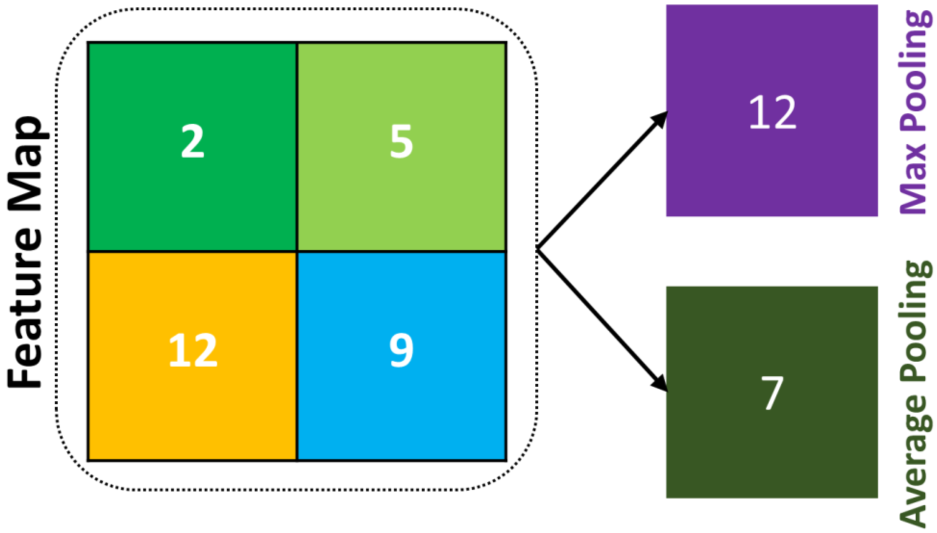


Fig. 2.2: Graphical representation of different types of pooling operation

Then to avoid the fitting problem of CNN, it is necessary to convert the feature map into a nonlinear one. This can be done by several nonlinear activation functions such as ReLU, sigmoid, tanh, etc. To get a human-like experience result ReLU is most commonly used. It eliminates the negative value from the feature map by replacing it with zero. ReLU is also capable of fastening the training procedure as it does not need complex algorithms and performs element-wise operations. As it returns zero for any negative input and does nothing for any positive input value, the ReLU activation function can be expressed as [65]-[66]:

$$f(x) = \max(0, x) \tag{2.3}$$

A basic operation of ReLU is shown in Fig. 2.3.

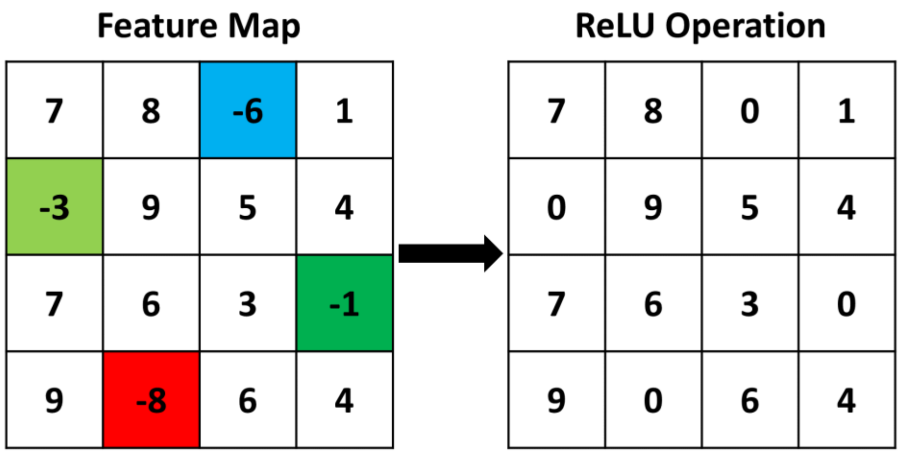


Fig. 2.3: Mechanism of ReLU function in Feature Map

Though pooling operation down samples, the image in respect of width and height; depth, or the quantity of channels always stays the same. However, to influence every output of the production vector by the input of the input vector, the fully connected layer is used by CNN models which is simply taught as a feed-forward neural system. The input of this layer is the output from the final pooling as a series.

The last layer of a neural network distributes and acknowledges the concluding output classifications of the model.

### 2.2.2 Dilated CNN

The working principle of Dilated CNN model is to insert a set of holes into the traditional convolutional layer thus growing the receptive field. The dilated convolution in a one-dimensional region can be defined [67]-[68] as:

$$o[i] = \sum_{l=1}^{L} f[i + r.l]k[l] \qquad (2.4)$$

Where, the function $f[i]$ represents input signal, $o[i]$ denotes the output signal value. To sample the function $f[i]$ the required dilation rates are indicated by the value of $r$. Moreover, the filter length of $L$, is expressed by the value of $k[l]$. From the above equation, traditional CNN and Dilated CNN can easily be differentiated as in traditional convolution the value of the dilation rate $r$ is always 1 and cannot be 1 in for a dilated convolution model.

The procedure can be simplified if $F : \mathbb{Z}^2 \rightarrow \mathbb{R}$ be a discrete function. Let $\sigma_r = [-r, r]^2 \cap \mathbb{Z}^2$ and if $k : \sigma_r \rightarrow \mathbb{R}$ be a discrete filter of size $(2r + 1)^2$. The discrete convolution operator $*$ can be expressed as

$$(F * k)(q) = \sum_{s + t = q} F(s)\, k(t). \tag{2.5}$$

The operator, $l$ is considered as a dilation factor and the value of $*_l$ can be expressed as

$$(F *_l k)(q) = \sum_{s + t = q} F(s)\, k(t). \tag{2.6}$$

Here, $*_l$ is represented as a dilated convolution. The accustomed distinct convolution * operator is merely the 1-dilated convolution.

Afterward to form the system considering $F_{0,} F_{1,} F_2, \dots, F_{n-1} : \mathbb{Z}^2 \rightarrow \mathbb{R}$ is discrete functions and considering $k_{0,} k_{1,} k_{2,} \dots, k_{n-2} : \sigma_r \rightarrow \mathbb{R}$ is distinct $3 \times 3$ filters. Thus, the whole filter operation is exponentially increasing dilation:

$$F_{i+1} = F_i *_{2^i} k_i \quad where\ i = 0, 1, 2, \dots, n - 2. \tag{2.7}$$

In the equation, the RF for the element $q$ in $F_{i+1}$, considers the set of elements in $F_0$ which change the value of $F_{i+1}(q)$. The number of these elements should equal the size of the receptive field of $q$ in $F_{i+1}$. Each element in $F_{i+1}$ has a receptive field that is $\left(2^{i+2} - 1\right) \times \left(2^{i+2} - 1\right)$ squared in size. The size of the receptive field is a square that grows exponentially [67]-[68].

### 2.2.3 Hybrid Dilated CNN

The basic principle of the HDC structure is to cover an input image using a filter without having any missing data or pixels at the final size of the Receptive Field [69]-[72]. Here Fig. 2.4 expresses the basic filter operation procedure of an HDC network along with traditional CNN.

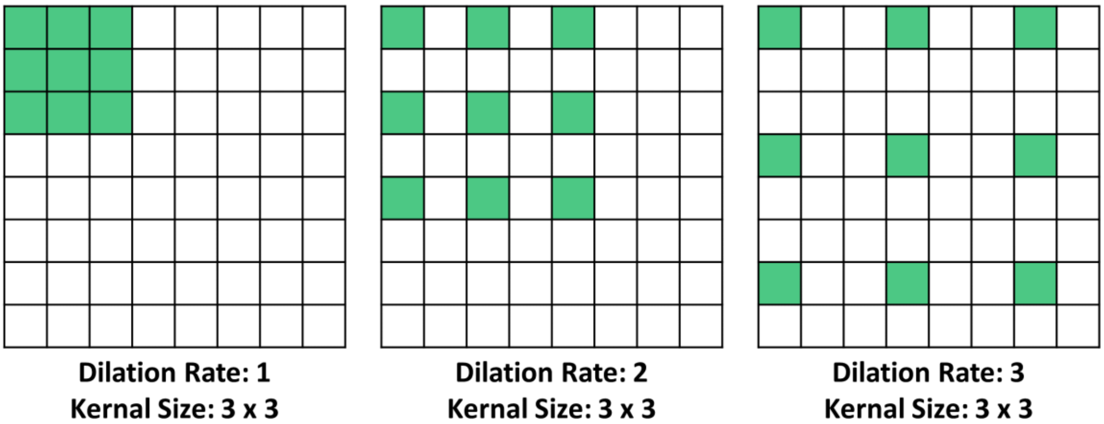


Fig. 2.4: Visualization of different dilation rate operations in the convolutional layer

From the above Fig. 2.4, it can be observed that traditional CNN has a filter size of $3 \times 3$ and dilation rate, $r = 1$, is producing a $3 \times 3$ receptive field. On the other hand, using the same filter size and utilizing a dilation rate of 2 can generate a $5 \times 5$ receptive field, and increasing the dilation rate to 3 can provide a $7 \times 7$ receptive field. Using this calculation process by manipulating the value of the dilation rate, a perfect HDC model can be designed to surpass the limitations of traditional CNN and Dilated CNN models. A basic design of the convolution calculation process is shown in Fig. 2.5.

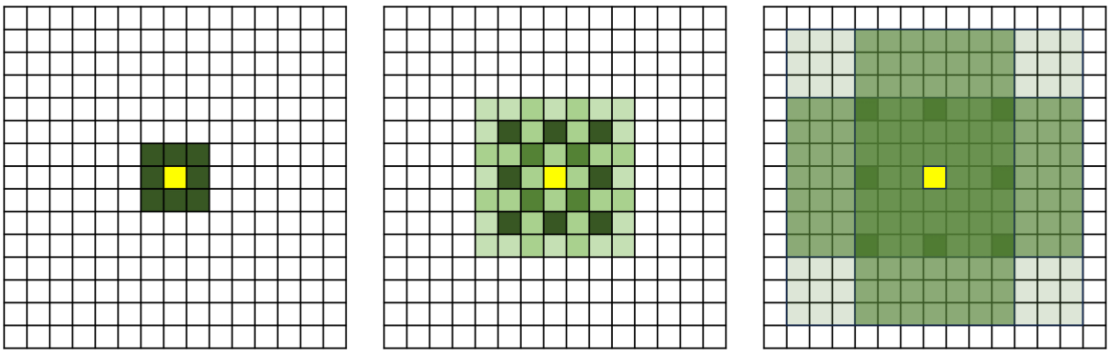

Fig. 2.5: Illustrated convolutional calculation process of the proposed HDC model

Here to reduce the griding effect and to eliminate the large distance irrelevant feature map a basic HDC kernel operation based on three different dilation rates, $r = 1, 2, 3$ on a single image is shown in Fig. 2.5. Here normal convolution is stacked by hybrid dilated convolution. The first layer is doing its operation having $r$ = 1. If any holes remain by the first layer, 2nd layer having $r$ = 2 can cover that as its operation starts right after the pixel used by 1st layer. Then 3rd layer having $r$ = 3 starts its feature mapping operation right after starting from the previous layer. Thus, these three layers are successfully covering a 13 * 13 size area without having any holes or missing pixels. It can simply be expressed that if we have an N convolutional layer with kernel size $K \times K$ having different dilation rates, the target is to cover the full input image region without giving any missing edges or holes.

## 2.3 CLASSIFIERS

One of the vital foundations in this study is the use of a classifier in the classification task of handwritten Bangla and English digits. Extracted features from dataset images are fed into classifiers for the final task. Renowned noteworthy classifiers are evaluated in this research with proper tuning. These classifiers are SoftMax, SVM, and KNN. The technical minutiae of the three classifiers are demarcated in this section.

### 2.3.1 SoftMax Classifier

To get the probability dispersal of classes, SoftMax is the vital layer of CNN architecture [73]-[74]. SoftMax assigns decimal probabilities in a multi-class problem to each class. It is implemented in multi-class cataloguing difficulties where class affiliation is required on more than two class labels. Just before the output layer SoftMax is implanted through a neural network. For both SoftMax and the output layer, the equal number of nodes must be maintained. The SoftMax function can be articulated as:

$$\sigma(\vec{Z})_i = \frac{e^{Z_i}}{\sum_{j=1}^{K} e^{Z_j}} \tag{2.8}$$

Here, $\vec{z}$ indicates the input vector. All the $z_i$ values are the components of the input vector and can yield any real value. The function $e^{z_i}$ is applied to all components of the input vectors. $\sum_{j=1}^{K} e^{z_j}$ function express normalization which ensures all the output values of the function is ended up to 1 and in the range of 0 to 1, which develops an authentic probability distribution. In this normalization function $K$ indicates the quantity of classes in the multi-class classifier.

### 2.3.2 SVM Classifier

In the arena of supervised learning systems, SVM is widely utilized for classification and also for regression problems. Mostly, in the field of machine learning, SVM is utilized for classification tasks. The SVM algorithm works on creating the perfect line or pronouncement boundary which can separate n-dimensional space into categories or classes. Which later helps the algorithm to decide on newly detected data to put in the proper category. This perfect line or decision-making boundary is mentioned as a hyperplane. SVM selects the extreme vectors and points to assist in the creation of the hyperplane. Support vectors, which are used to represent these extreme instances, are the foundation of the SVM technique [75]-[77]. For two different categories to crack the classification problem, the procedure of SVM is shown in Fig. 2.6 with proper labeling.

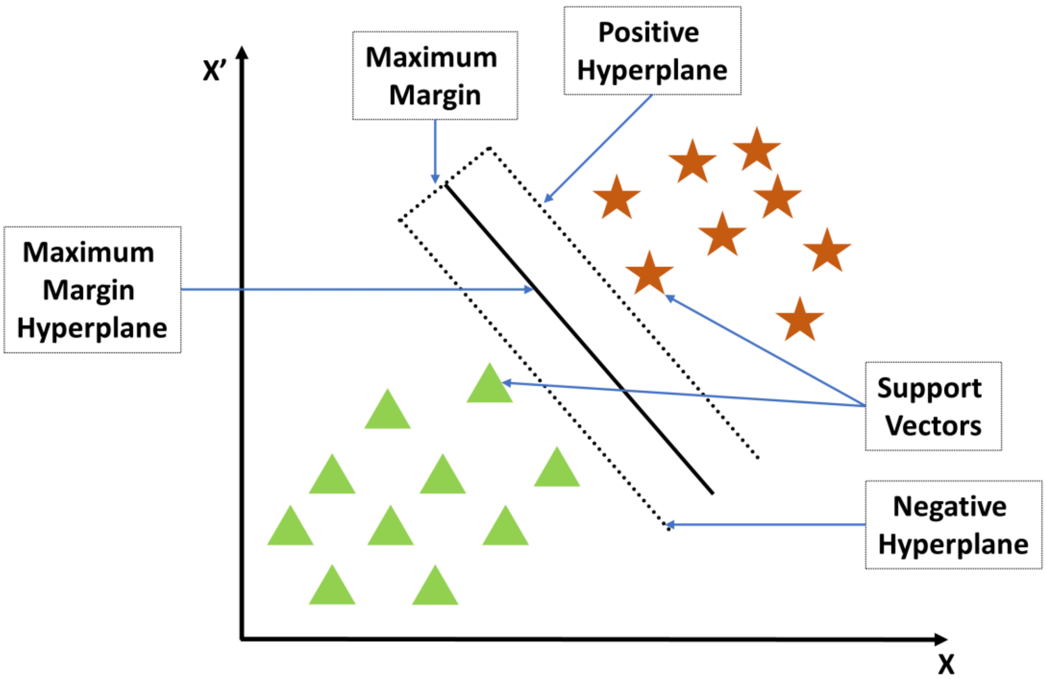


Fig. 2.6: Working technique of the SVM algorithm

Here in Fig. 2.6, it is seen that SVM performs classification creating hyperplane or decision boundary. Support vectors are the data points, or vectors in this case, that are closest to the hyperplane and have the greatest significance to its location. These vectors are referred to as support vectors due to how they aid the hyperplane.

### 2.3.3 KNN Classifier

One of the most convenient machine learning algorithms is K-Nearest Neighbor, which has its foundation on the approach of supervised instruction. Assuming that the new case and the existing cases are comparable, the KNN algorithm places the new instance in the category that most closely resembles the current categories. The KNN technique is used to classify a new data point based on similarity after all of the current data has been captured. This suggests that by using the KNN method, new data can be accurately and quickly sorted into the right category.

Although the KNN technique is most frequently employed to address classification challenges, it can also be applied to regression problems. Since KNN is a non-parametric technique, it makes no speculation about the underlying data. This algorithm is also known as an idle learner since it saves the training information rather than immediately using it for learning. Instead, it manipulates the dataset to classify the information.

The KNN approach only provisions the information during the training stage and assigns it to a category that closely resembles the newly received data [78]-[81]. A basic operation principle of KNN is shown in Fig. 2.7.

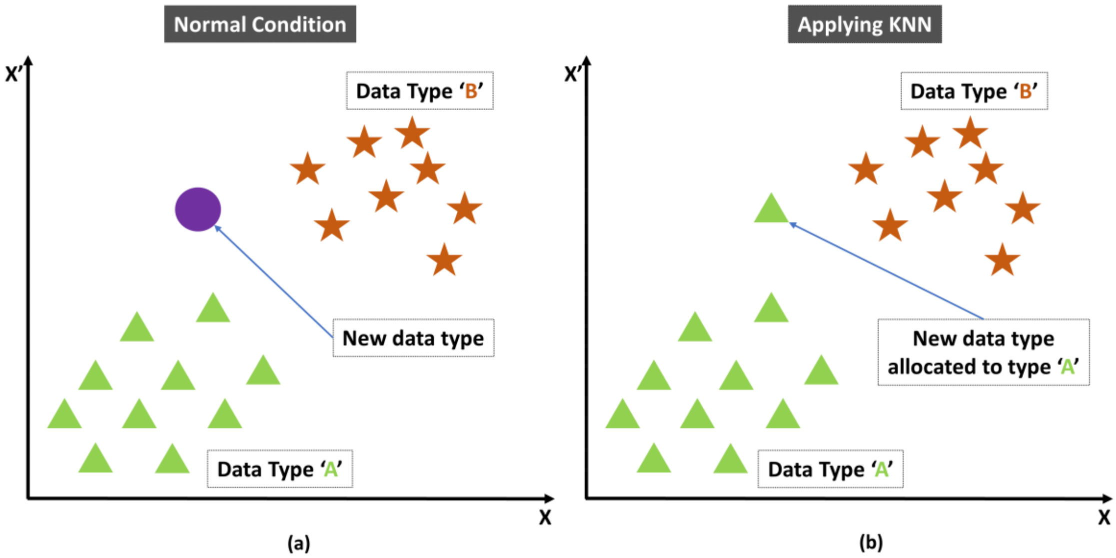


Fig. 2.7: Illustrated working principle of the KNN algorithm

From Fig. 2.7 it can be observed that, if there are two data types such as Data Type 'A' and Data Type 'B' and there arrives a new data type between axis then applying KNN based on Euclidian distance calculation the new data type is identifiable and is allocated to the particular data type.

# Chapter 3: METHODOLOGY AND MATERIALS

*Methodology, Research Design, Dataset, Dataset Pre-processing, and Design & Development of the Proposed Models form the basis of this chapter. This chapter describes the entire process of this research, including the dataset used for justification and dataset improvements. Moreover, all the proposed five models are expressed elaborately in this chapter mentioning details of the design and development procedure along with simulation data. Mathematical Parameter Evaluation of the first three models is also shown in this chapter.*

## 3.1 METHODOLOGY AND RESEARCH DESIGN

The primary component of several traditional Bangla digit classification methods is the convolution neural network (CNN). The classic CNN has two major problems: its computations are too resource-intensive and require more simulation time [82]. To overcome these problems, at first, Dilated CNN [67] is applied in this research to classify handwritten Bangla digits. The Dilated CNN faces accuracy loss as it loses pixels due to its convolution procedure.

Then, a Hybrid Dilated CNN model is proposed by using dilated convolution kernels with diverse dilated rates to eliminate the detail loss problem of the dilated CNN model. The proposed Hybrid Dilated CNN model can achieve larger receptive fields and can extract distant features.

Finally, support vector machine, and k-nearest neighbor are utilized as a classifier to increase the classification accuracy. This work focuses on using less training data, less training time, and fewer computing resources but a highly efficient classification structure. To shape up the above-mentioned outline this research follows a systematic procedure by developing several models. The working procedure of this proposed system is expressed in Fig. 3.1.

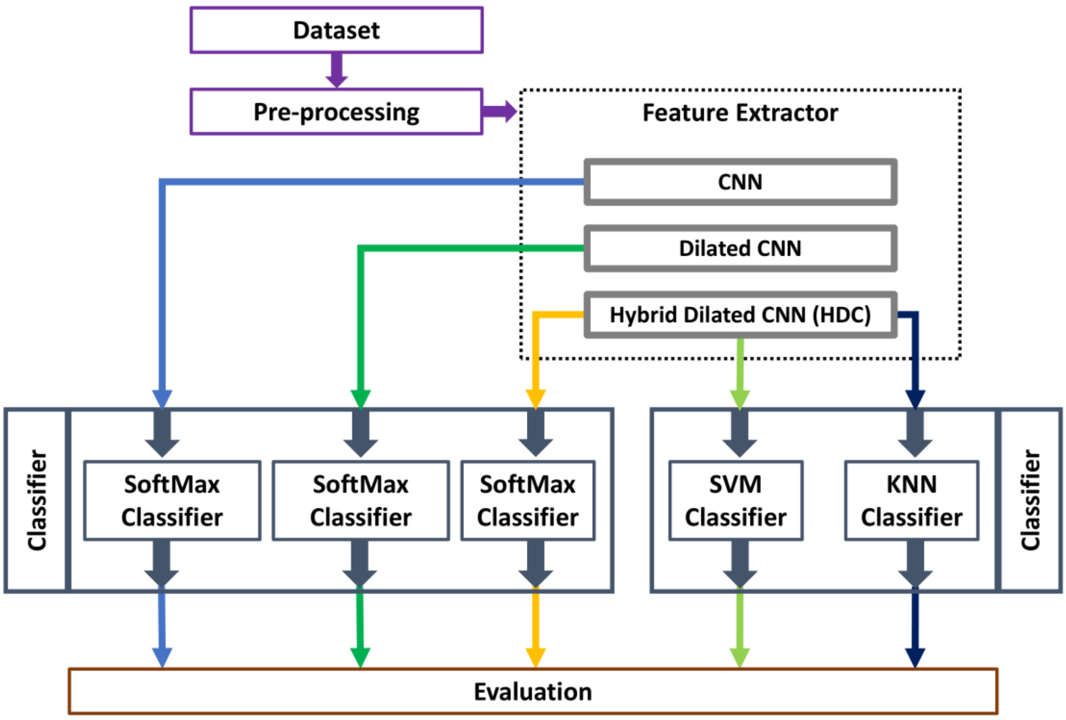

Fig. 3.1: Working procedure of the proposed method

In this work, a special CNN model is first created. Which, in terms of user level and industrial uses and in standings of time management, has significant problems recognizing handwritten digits. Then in this research, a Dilated CNN model is suggested as a solution to CNN's drawbacks. While this Dilated CNN model resolves CNN's time consumption issues, it introduces additional accuracy issues by pressing holes into the feature map.

In this study, an innovative Hybrid Dilated CNN Model (HDC) is built after considering the issues that come from both CNN and Dilated CNN. This performs admirably and eventually overcomes limitations imposed by the CNN and Dilated CNN models. Finally, this study implements two well-known machine learning algorithms, SVM and KNN, for classification tasks using features provided by Hybrid Dilated CNN Model (HDC) as input.

## 3.2 DATASETS

Two different handwritten digit dataset are used in this research. In the case of the Bangla handwritten digit recognition segment, the 'BanglaLekha-Isolated' dataset is used. On the other hand, in the scenario of the English digit recognition task, the 'MNIST' dataset is utilized. Both datasets are demarcated in section 3.2.1 and section 3.2.2 in this chapter.

### 3.2.1 BanglaLekha-Isolated Dataset

BanglaLekha-Isolated [83] dataset is used in this work for Bangla digit recognition. This dataset is enriched with digits and characters of Bangla handwritten from diverse topographical locations across Bangladesh including a variety of writer groups. This work uses Bangla digits from '0 (০)' to '9 (৯)' from this dataset and re-labeled in the English Language for recognition. A representative sample of all Bangla digits acquired from this dataset in this work is shown in Fig. 3.2 with re-labeling information. For all ten Bangla digits, 19000-digit data are taken in this work consisting of 1900 handwritten Bangla digits per class.


Fig. 3.2: Handwritten Bangla digit dataset samples

### 3.2.2 MNIST Dataset

MNIST [84] dataset is used in this work for handwritten English digit appreciation. The Modified National Institute of Standards and Technology database, or MNIST dataset, is a great collection of handwritten numbers that is frequently utilized to train innumerable image processing systems. The dataset is also broadly used for testing and training in the machine learning space. It was shaped by "re-mixing" the initial illustrations from the NIST dataset. The dataset's black-and-white images were also adjusted to fit into a 28 by 28 pixel bounding box, anti-aliased, and given grayscale echelons. This dataset is enriched with handwritten English from diverse topographical locations including a variety of writer groups. This work uses English digits from ‘0’ to ‘9’ from this dataset and labels in the English Language for recognition. A representative sample of all English digits acquired from this dataset in this work is shown in Fig. 3.3 with labeling information. For all ten English digits, 60000-digit data are taken in this work consisting of 6000 handwritten Bangla digits per class.


Fig. 3.3: Handwritten English digit dataset samples

### 3.3 DATASET PRE-PROCESSING

These datasets are efficiently pre-processed [83]-[84]. Firstly, two have a black background with the letter written in white, the background and foreground are inverted. Then the median filter is utilized to take away the unnecessary noises. For better edge detection a filter relevant to edge thickening is utilized coherently. The digit samples resized the 60-to-60-pixel square rate for the BanglaLekha-Isolated dataset and the 28-to-28-pixel rate for the MNIST dataset by keeping the proper aspect ratio along with color correction for better training performance. An overview of the entire dataset pre-processing steps for both datasets is expressed in Fig. 3.4.

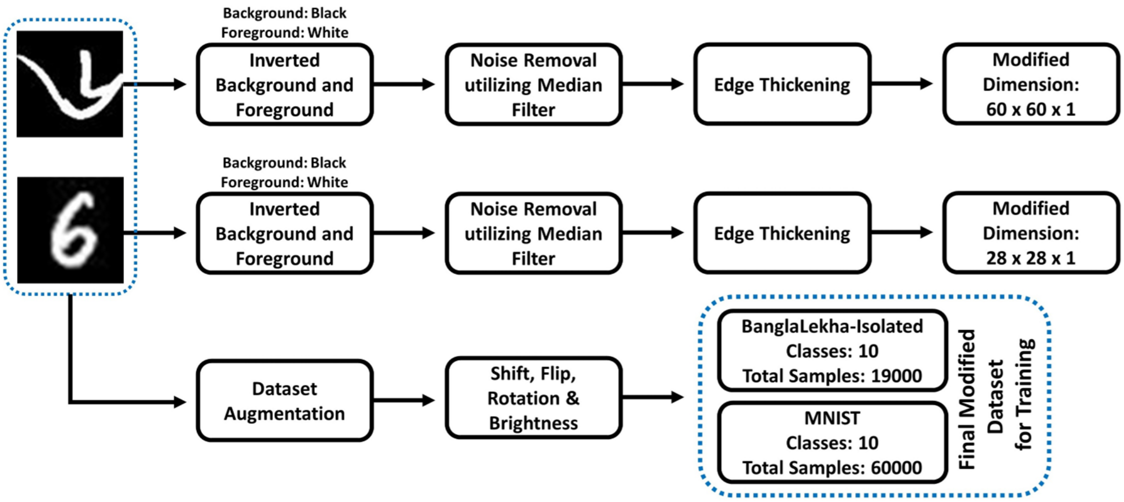


Fig. 3.4: Flow illustration of dataset pre-processing steps

All the pre-processing steps adopted in this work are briefly expressed in the following section with proper technical insights.

### 3.3.1 Inverted Background and Foreground

Inverted background and foreground techniques are utilized in the case of both datasets. The idea is to utilize the high-fidelity contrast of white color. Here, in this research, the targeted objects are converted into white color no matter in which color it was. The other regions of each dataset image are converted into black pixels. Thus, this technique increases the developed model's picture detailing capability as less area needs to be attended during the feature mapping procedure.

### 3.3.2 Noise Removal

One of the significant pre-processing methods adopted in this work for dataset preprocessing is the use of a median filter for noise removal. The median filter is a non-linear electronic filtration strategy that frequently gets utilized to remove noise from a signal or image. Noise reduction is a prominent pre-processing method for boosting the results of the subsequent processing. For the reason that it can preserve edges while plummeting noise, median filtering is a particularly prevalent technique in numerical image processing. It also has uses in signal processing. The central function of the median filter is to exchange each entry in the signal, entry by entry, with the median of the entry and the admissions that surround it. The concept is quite similar to that of a moving average filter, in which the arithmetic mean of each item and its neighbors is used in place of each entry. The "window" states to the neighbors' outline that moves throughout the signal, entry by entry. The furthermost obvious window for one-dimensional signals is simply the first few entries that come before and after; however, for two-dimensional (or higher-dimensional) data, the window requests to comprehend all entries that fall inside a specific radius, ellipsoidal, or rectangular section [85]. Utilizing this median filter in this work's pre-processing steps makes the system fluent and flawless.

### 3.3.3 Edge Thickening

Another major step taken in the pre-processing steps is to thicken the edges of the dataset images. Because thickening the edges contributes to the relativity in adjacent pixels and supports more relevant feature mapping.

The Frangi vesselness sorting is used by the fibermetric filter to amplify elongated or tubular objects in the image. The fibermetric filter uses a multiscale Frangi vesselness filter with a Hessian basis to emphasize elongated or canular assemblies in a 2-D or 3-D grayscale image. The image that is returned has the filter's highest response at a thickness that roughly corresponds to the tubular structure's dimensions. The thickness of the tubular structures to be improved is specified, and other parts of the filtering method are controlled by name-value pair inputs. There is no segmentation carried out by the fibermetric filter. The function, which is usually used as a preprocessing step for segmentation, highlights structures in an image [86].

Thus, considering all the pros of fiber metric filters, in the case of edge thickening of dataset images, this technology is utilized in this research.

### 3.3.4 Image Resize

As the raw dataset images are in diverse proportions and profiles, it is mandatory to arrange and shape them perfectly to train the developed model efficiently and fluently. Besides, this technique reduces the training time at a large scale. So, both datasets in this work are modified from raw format and an acceptable shape is given. To do the task MATLAB's "imresize" function is utilized in this work.

In the scenario of MATLAB's "imresize" function, the result of X = imresize(Y, scale) is image X multiplied by the size of picture Y. Grayscale, RGB, binary, or category images can all be used as the input image Y. Imresize only adjusts the first two dimensions of Y if it has more than two. X is smaller than Y if the scale falls between 0 and 1. X is larger than Y if the scale is greater than 1. Imresize employs bicubic interpolation by default. The image X with the quantity of rows and columns indicated by the two-element vector [numrows numcols] is returned by the expression X = imresize(Y,[numrows numcols]) [87].

Utilizing the above-mentioned technique the images of the "BanglaLekha-Isolated" and "MNIST" datasets are modified 60 by 60 pixel, and 28 by 28 pixel respectively.

## 3.4 FEATURE EXTRACTION

One of the major parts and also significant contributions of this research is in the segment of feature extraction. This research tries to develop a state-of-the-art scheme for extracting features from handwritten digit datasets. Convolutional Neural Network (CNN), Dilated CNN, and Hybrid Dilated CNN algorithm's feature mapping technique are evaluated in this research for a noteworthy solution. The basic principles of these algorithms are discussed briefly in Chapter 2.

## 3.5 CLASSIFIER EVALUATION AND ANALYSIS

Another major contribution of this research is in the segment of classification. Five different full pledger models are developed in this research. Here, CNN, Dilated CNN, and Hybrid Dilated CNN (HDC)'s convolutional technique is utilized in this work for feature extraction from the dataset. Then those extracted features are fed into different classifiers. SoftMax, SVM, and KNN classifiers are utilized in this work for classification tasks. All pros and cons, and brief comparisons of these classifiers are explored in this work with proper mathematical calculation and simulation.

## 3.6 CNN MODEL

Keeping the above layer working principle mentioned in Chapter 2 [60]-[66], the CNN model of this work consists of a repeated six-group batch of 2-dimensional convolutional layer, batch normalization layer, ReLU layer, and max pooling layer.

### 3.6.1 Design of CNN Model

A detailed report about layer name, layer type, number of filters, and filter size is specified below in Table 3.1.

Table 3.1: Layer analysis of the CNN model

| **Layer Name** | **Layer Type** | **Activations** | **Number of Filter** | **Filter Size** | **Number of Learnable** |
|---|---|---|---|---|---|
| Input | Image Input | 60(S) x 60(S) x 3(C) x 1(B) | | | 0 |
| 1st Conv | Convolution | 60(S) x 60(S) x 8(C) x 1(B) | 8 | 3*3 | 224 |
| 1st BN | Batch Normalization | 60(S) x 60(S) x 8(C) x 1(B) | | | 16 |
| 1st Relu | ReLU | 60(S) x 60(S) x 8(C) x 1(B) | | | 0 |
| 1st MaxPool | Max Pooling | 30(S) x 30(S) x 8(C) x 1(B) | | | 0 |
| 2nd Conv | Convolution | 30(S) x 30(S) x 16(C) x 1(B) | 16 | 3*3 | 1168 |
| 2nd BN | Batch Normalization | 30(S) x 30(S) x 16(C) x 1(B) | | | 32 |
| 2nd Relu | ReLU | 30(S) x 30(S) x 16(C) x 1(B) | | | 0 |
| 2nd MaxPool | Max Pooling | 15(S) x 15(S) x 16(C) x 1(B) | | | 0 |
| 3rd Conv | Convolution | 15(S) x 15(S) x 32(C) x 1(B) | 32 | 3*3 | 4640 |
| 3rd BN | Batch Normalization | 15(S) x 15(S) x 32(C) x 1(B) | | | 64 |
| 3rd Relu | ReLU | 15(S) x 15(S) x 32(C) x 1(B) | | | 0 |
| 3rd MaxPool | Max Pooling | 7(S) x 7(S) x 32(C) x 1(B) | | | 0 |
| 4th Conv | Convolution | 7(S) x 7(S) x 64(C) x 1(B) | 64 | 3*3 | 18496 |
| 4th BN | Batch Normalization | 7(S) x 7(S) x 64(C) x 1(B) | | | 128 |
| 4th Relu | ReLU | 7(S) x 7(S) x 64(C) x 1(B) | | | 0 |
| 4th MaxPool | Max Pooling | 3(S) x 3(S) x 64(C) x 1(B) | | | 0 |
| 5th Conv | Convolution | 3(S) x 3(S) x 128(C) x 1(B) | 128 | 3*3 | 73856 |
| 5th BN | Batch Normalization | 3(S) x 3(S) x 128(C) x 1(B) | | | 256 |
| 5th Relu | ReLU | 3(S) x 3(S) x 128(C) x 1(B) | | | 0 |
| 5th MaxPool | Max Pooling | 1(S) x 1(S) x 128(C) x 1(B) | | | 0 |
| 6th Conv | Convolution | 1(S) x 1(S) x 256(C) x 1(B) | 256 | 3*3 | 295168 |
| 6th BN | Batch Normalization | 1(S) x 1(S) x 256(C) x 1(B) | | | 512 |
| 6th Relu | ReLU | 1(S) x 1(S) x 256(C) x 1(B) | | | 0 |
| FC | Fully Connected | 1(S) x 1(S) x 10(C) x 1(B) | | | 2570 |
| SoftMax | SoftMax | 1(S) x 1(S) x 10(C) x 1(B) | | | 0 |
| Output Classification | Classification Output | 1(S) x 1(S) x 10(C) x 1(B) | | | 0 |

### 3.6.2 Training and Validation of CNN Model

Training parameters specified in this work are shown in Table 3.2. The Stochastic Gradient Descent with Momentum (SGDM) solver is utilized in the projected CNN model.

Table 3.2: Training specification of the CNN model

| Training Options | Values |
|---|---|
| Initial Learn Rate | 0.01 |
| Solver | sgdm |
| Shuffle | every-epoch |
| Max Epoch | 6 |
| Validation Frequency | 30 |
| Validation Data | Test Images |
| Verbose | false |

The performance and learning-related parameters of the developed CNN structure are given below in Table 3.3.

Table 3.3: Validation report of the developed CNN model

| Parameters | Values |
|---|---|
| Accuracy | 96.48 % |
| Time required | 63 seconds |
| Iteration | 666 of 666 |
| Epoch | 6 of 6 |
| Iterations per epoch | 111 |
| Maximum iterations | 666 |
| Validation Frequency | 30 iterations |

The proposed CNN model shows a 96.48% accuracy and consumes 63 seconds to perform the whole task. To get a proper overview of the model; precision, recall, and f1 score are premeditated and expressed in Fig. 3.5 for better evaluation.

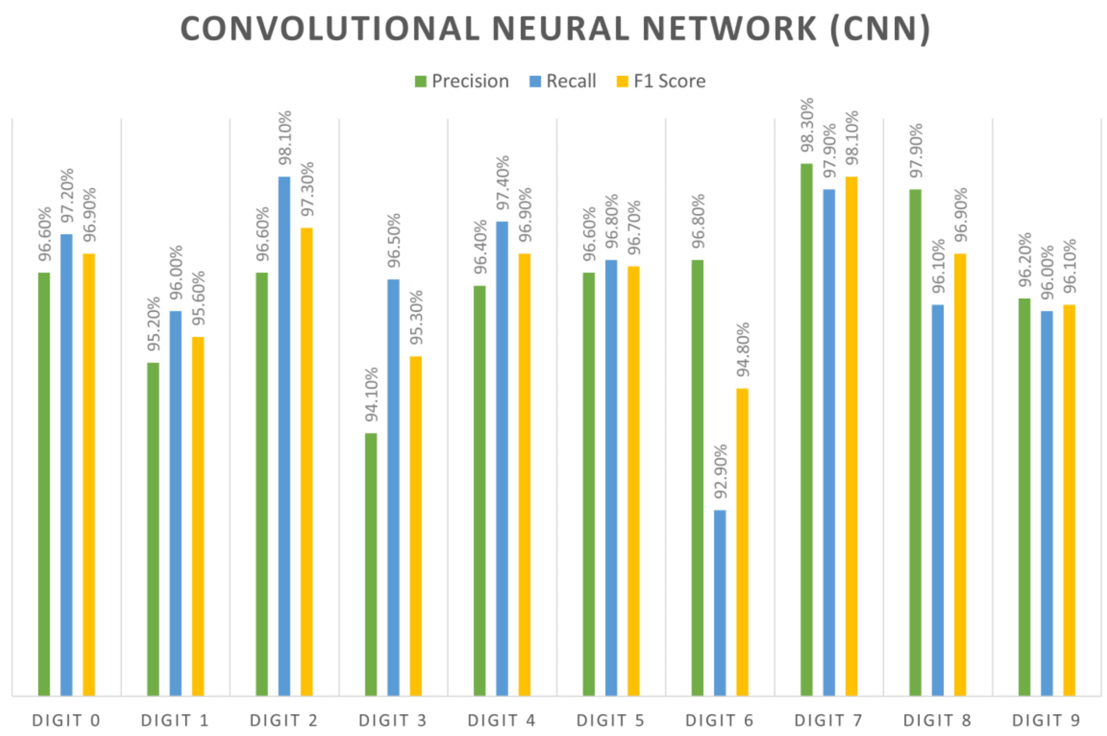


Fig. 3.5: Precision, Recall, and F1 Score evaluation graph for all categories in the CNN model

## 3.7 DILATED CNN MODEL

In the CNN model, several data loss occurs due to the skipping of pixel points in the feature map for the operation criteria of the pooling technique. This affects the final evaluation process of the feature map. Due to this pixel loss effect sometimes, accuracy is not up to the mark. Besides this, the CNN model's convolution process covers a smaller receptive field area, which needs more steps to develop a feature map so it requires more training time and more computing power. In addition, regular standard CNN focuses on repetitive operations and ignores the spatial similarity of adjacent regions. So, this work focuses on these problems to solve and make an efficient recognition model.

### 3.7.1 Design of Dilated CNN Model

In this work, the convolution kernels of CNN are replaced by the dilated convolution kernels to overcome the drawbacks of the conventional CNN model. A modified Dilated CNN is developed based on the original CNN model for handwritten Bangla digit recognition.

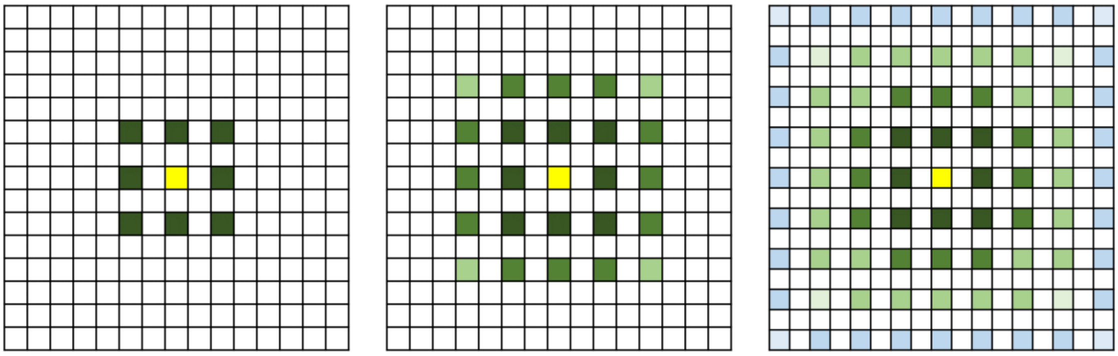


Fig. 3.6: Convolutional operation of the Dilated CNN model

The operating principle of this proposed Dilated CNN's convolutional operation is shown in Fig. 3.6. The model is formed by replacing the traditional convolutional kernel. Here, the dilation rate, $r$ is 2 and the kernel size is $3 \times 3$. In normal convolution, it covers a $3 \times 3$ receptive field for normal dilation, but due to the dilation rate $r$ = 2, the $3 \times 3$ filters cover a $5 \times 5$ receptive field area, shown in Fig. 3.6 (a). For the next step, the receptive field increases to $9 \times 9$ which is $5 \times 5$ in regular convolution shown in Fig. 3.6 (b). Fig. 3.6 (c) presents RF for the third convolution step. By following the order using a $3 \times 3$ size filter with dilation rate 2, in this work, covers $17 \times 17$ pixel RF areas. From Fig. 3.6, it can easily be observed that Dilated CNN can cover a larger receptive field area than traditional CNN consuming the same amount of time and having a large vision in feature analysis due to extended area size in a single time. Theoretically, a fixed dilation rate is set to all convolution operations in a Dilated CNN model by resetting the stride value to 1 to remove down sampling [56], [67]-[68]. To calculate the dilated filter for a convolution kernel with a size of $k \times k$, the size of the final dilated filter is $k_{df} \times k_{df}$.

Here,

$$k_{df} = k + (k-1).(r-1). \qquad (3.1)$$

If the kernel size $k$ =3, and dilation rate $r$ =2, then the value of $k_{df} = 3 + (3-1).(2-1) = 5$. Now for the first dilated convolution operation, for a fixed $3 \times 3$ filter having dilation rate, r =2 the dilated filter is $5 \times 5$.

For the next dilated convolution operation, the value of $k$ changed to 5. Now, again utilizing equation 7, $k_{df} = 5 + (5-1).(2-1) = 9$ resulting in a new dilated filter producing a size of $9 \times 9$.

For the next dilated convolution operation, the value of $k$ changed to 9. Now, again utilizing equation 7, $k_{df} = 9 + (9-1).(2-1) = 17$ resulting in a new dilated filter producing a size of $17 \times 17$.

From the above analysis, it can be observed that the Dilated CNN's convolutional operation covers a larger RF area than the normal CNN's convolutional operation consuming same number of steps. For that, Dilated CNN can cover more features than traditional CNN and consumes less training time. In traditional CNN the dilation rate is 1 and in Dilated CNN, the dilation is greater than 1 and is the same for all convolutional layers. The developed dilated model in this work is expressed in Fig. 3.7.

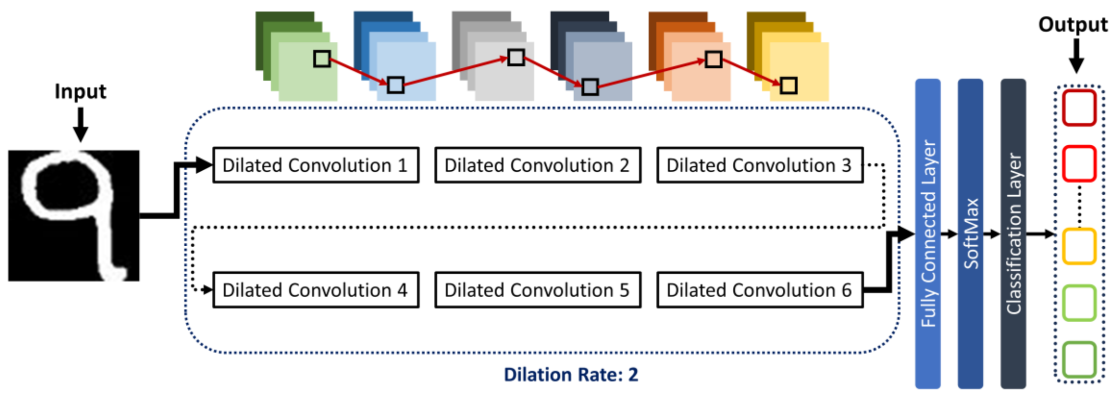


Fig. 3.7: Block illustration of the developed Dilated CNN model

### 3.7.2 Training and Validation of Dilated CNN Model

This model is designed on the base of the developed CNN structure in this work. Remaining the parameters are equivalent and a dilation rate, $r$ = 2 is applied to all six convolution operations for better observation of the difference between the two models. Here Table 3.4 shows the performance analysis of the developed Dilated CNN structure with a comparison to the CNN structure.

Table 3.4: Comparison of validation parameters of the CNN and Dilated CNN model

| Parameters | Values (Dilated CNN model) | Values (CNN Model) |
|---|---|---|
| Accuracy | 95.85 % | 96.48 % |
| Time required | 52 seconds | 63 seconds |
| Epoch | 6 of 6 | 6 of 6 |
| Iteration | 666 of 666 | 666 of 666 |
| Iterations per epoch | 111 | 111 |
| Maximum iterations | 666 | 666 |
| Validation Frequency | 30 iterations. | 30 iterations |

From the above table, it can be observed that using equivalent parameters Dilated CNN model takes 11 seconds less time by using fewer computing resources. This is the foremost goal of this work to reduce computing time for industry and user-level recognition systems. But, 0.63% accuracy is lost in this process. The reason for accuracy loss is that, as Dilated CNN inserted holes to get an extended feature map to cover a larger receptive field it inserted holes weighted zero. It reduces training time and uses computing resources significantly along with making some pixel data loss in every step. These feature maps are less clear in comparison to traditional CNN.

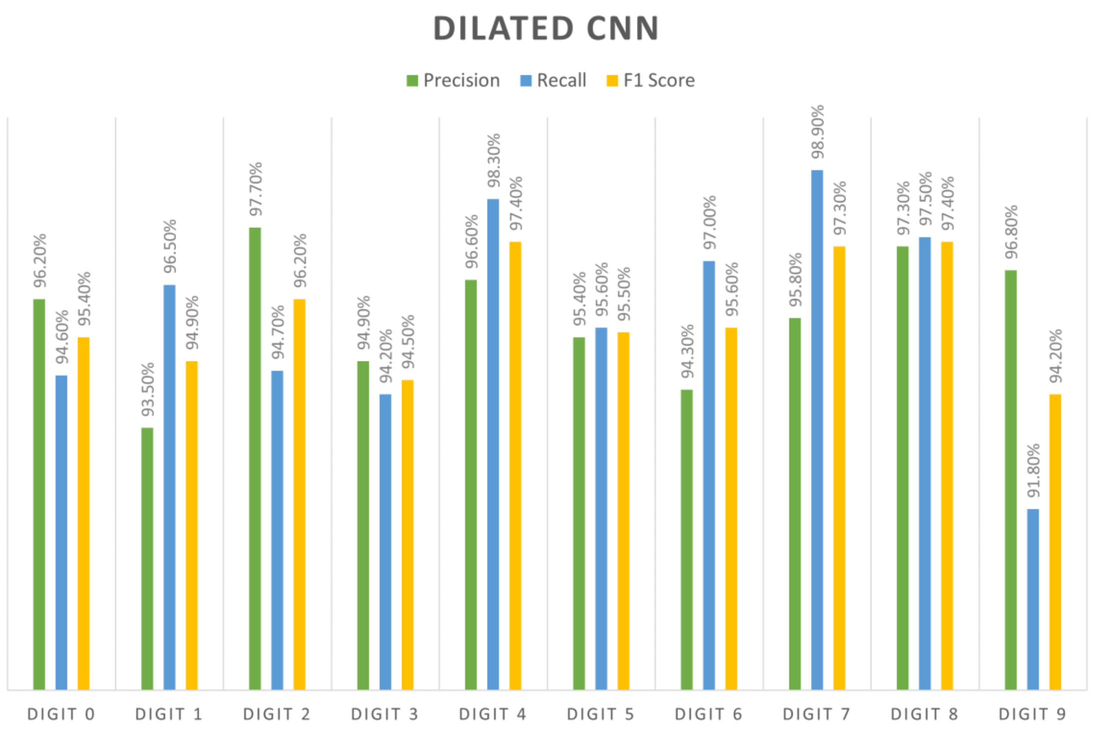


Fig. 3.8: Precision, Recall, and F1 Score evaluation graph for all categories in the Dilated CNN model

To observe those drawbacks of the projected Dilated CNN model, and also to evaluate effects in every category, the precision, recall, and f1 score are calculated for all the classes and expressed in Fig. 3.8.

### 3.7.3 Limitations of Dilated CNN Model

From the above Dilated CNN module two significant problems can be identified, the first one is that the kernel is not continuous; which simply is known as 'gridding'. The other problem is having inserted zero-value pixels in a large information map, which sometimes can appear inappropriate for smaller objects [88]-[90]. From the above convolution operation in Fig. 3.6, it can be observed that for a pixel value in a dilated convolutional layer l, the data that feeds in that pixel value, generates from the adjacent $k_{df} \times k_{df}$ region. As dilated convolution layers increase the filter size increases by inserting holes i.e. the actual data is stored from a lesser filter size which is causing an information gap. As the gap is $r - 1$ in between two non-zero pixels, if the filter size is 3 and the dilation rate is 2 only 9 out of 25 pixels in the receptive field are utilized for feature calculation. If the dilation rate increases more and as the dilation rate is set in fixed value in dilated operation, more local data is going to be lost. For only dilation rate 2 at least 75% image portion can be lost [90, 91]. Besides taking data to generate a feature map skipping middle pixels in several series can create improper dataset features, thus fells effect in accuracy.

## 3.8 HYBRID DILATED CNN MODEL

From the previous observation of this work, one significant improvement and two major drawbacks are introduced. These are (i) The mechanism of dilated convolution operation can reduce training time and computing resources consumption at a large scale however, (ii) larger values of dilation rate having repeated convolutional operation makes a huge amount of local information loss and (iii) due to inserted holes having zero value in expanded filter size cause irrelevant large distance feature map.

To solve the problem mentioned above, a unique Hybrid Dilated CNN (HDC) model is proposed in this work.

### 3.8.1 Design of Hybrid Dilated CNN Model

Besides solving the above-described problem Hybrid Dilated CNN (HDC) is utilized in this work as it has superior feature-extracting capability rather than CNN. Which is in the segment of hyperspectral images and has already been proven by many previous works [56], [82], [91]-[92]. In comparison to hyperspectral image characteristics, the writing style of handwritten digits also shares some common but significant features that cannot easily be obtained by CNN but HDC is proven to do it. Those specific digit features are moment-based features such as geometric moments, moment invariants, affine moment invariants, and zernike moments along with slope and curvature types.

As HDC is already proven to extract those features in the segment of Spatial analysis of hyperspectral images, this geometric feature extraction capability of HDC can crack more features from handwritten datasets rather than CNN.

To design an HDC model, the main concern point is the dilation rate in the layer of convolution. This strategy of applying different dilation rates in different convolutions differs HDC model from CNN and Dilated CNN. Before developing the proposed HDC model, this work evaluated previously done works in this segment to utilize all the prominent outcomes. Context aggregation model [67], Semantic segmentation model [91], Atrous spatial pyramid pooling model [92], HDCFE-Net [56], and Image Classification model [82] are deeply evaluated to design the proposed HDC model for better training time, more feature acquisition, and better classification results.

The first four layers of the proposed model are applied with different dilation rates, $r$ = 1, 2, 3, and 5 which satisfies the following expression proposed by [91].

$$Mi = \max[M_{i+1} - 2r_i, M_{i+1} - 2(M_{i+1} - r_i), r_i] \qquad (3.2)$$

Where, $r_i$ represents the $i_{th}$ layer's dilation rate. $M_i$ is the largest $r_i$ in the $i_{th}$ layer. The value of $i$ indicates 1, 2, 3, … , n. The objective of the layout is to let $M_2 \leq K$. If we consider the kernel size, $K = 3$, then following the above logic, an $r = [1, 2, 3, 5]$ pattern can be utilized.

The last two convolution layers have dilation rates $r$ = 9, 11 respectively inspired by the Atrous spatial pyramid pooling model [92] to cover the whole image as a flash view and to make capable the model detect spatial type features and recover every feature map processed by the dilation rates 1, 2, 3, and 5.

A block diagram of the flourished HDC model in this work is expressed in Fig. 3.9. The proposed model is equivalent in basic parameters with the original CNN model in this work for better justification of this work's contribution.

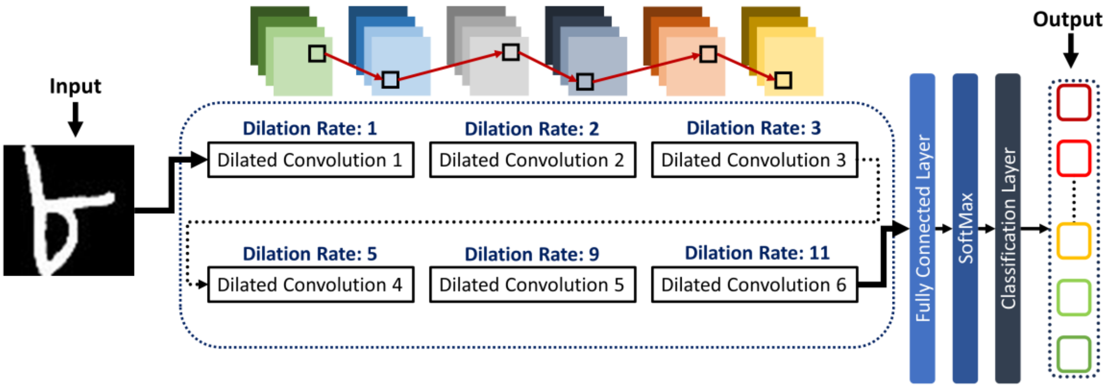


Fig. 3.9: Block illustration of the proposed HDC model

### 3.8.2 Training and Validation of Hybrid Dilated CNN Model

Training parameters kept equivalent to the proposed CNN and Dilated CNN for to-the-point evaluation in results. A detailed procedure of the performed training structure on the developed HDC model is expressed in Fig. 3.10.

Training on single GPU.
Initializing input data normalization.

| Epoch | Iteration | Time Elapsed (hh:mm:ss) | Mini-batch Accuracy | Validation Accuracy | Mini-batch Loss | Validation Loss | Base Learning Rate |
|---|---|---|---|---|---|---|---|
| 1 | 1 | 00:00:03 | 6.25% | 20.36% | 2.7685 | 2.2035 | 0.0100 |
| 1 | 30 | 00:00:05 | 89.06% | 89.52% | 0.3733 | 0.3377 | 0.0100 |
| 1 | 50 | 00:00:07 | 92.19% | | 0.2408 | | 0.0100 |
| 1 | 60 | 00:00:08 | 92.19% | 92.72% | 0.2422 | 0.2415 | 0.0100 |
| 1 | 90 | 00:00:10 | 88.28% | 94.32% | 0.3562 | 0.1934 | 0.0100 |
| 1 | 100 | 00:00:10 | 96.88% | | 0.1079 | | 0.0100 |
| 2 | 120 | 00:00:12 | 96.88% | 94.02% | 0.1180 | 0.1956 | 0.0100 |
| 2 | 150 | 00:00:14 | 94.53% | 94.86% | 0.1394 | 0.1699 | 0.0100 |
| 2 | 180 | 00:00:16 | 96.88% | 94.95% | 0.1196 | 0.1711 | 0.0100 |
| 2 | 200 | 00:00:17 | 93.75% | | 0.1164 | | 0.0100 |
| 2 | 210 | 00:00:18 | 96.09% | 95.12% | 0.1163 | 0.1610 | 0.0100 |
| 3 | 240 | 00:00:21 | 99.22% | 95.62% | 0.0605 | 0.1547 | 0.0100 |
| 3 | 250 | 00:00:21 | 98.44% | | 0.0782 | | 0.0100 |
| 3 | 270 | 00:00:23 | 93.75% | 95.62% | 0.2360 | 0.1511 | 0.0100 |
| 3 | 300 | 00:00:25 | 98.44% | 95.75% | 0.1550 | 0.1440 | 0.0100 |
| 3 | 330 | 00:00:28 | 94.53% | 95.66% | 0.1351 | 0.1461 | 0.0100 |
| 4 | 350 | 00:00:29 | 98.44% | | 0.0824 | | 0.0100 |
| 4 | 360 | 00:00:30 | 96.88% | 96.25% | 0.0972 | 0.1353 | 0.0100 |
| 4 | 390 | 00:00:32 | 97.66% | 96.02% | 0.0725 | 0.1310 | 0.0100 |
| 4 | 400 | 00:00:33 | 97.66% | | 0.0771 | | 0.0100 |
| 4 | 420 | 00:00:35 | 99.22% | 96.13% | 0.0329 | 0.1387 | 0.0100 |
| 5 | 450 | 00:00:37 | 98.44% | 96.17% | 0.0333 | 0.1364 | 0.0100 |
| 5 | 480 | 00:00:39 | 99.22% | 96.00% | 0.0220 | 0.1327 | 0.0100 |
| 5 | 500 | 00:00:40 | 99.22% | | 0.0324 | | 0.0100 |
| 5 | 510 | 00:00:42 | 100.00% | 96.23% | 0.0117 | 0.1305 | 0.0100 |
| 5 | 540 | 00:00:44 | 100.00% | 96.11% | 0.0168 | 0.1383 | 0.0100 |
| 5 | 550 | 00:00:44 | 100.00% | | 0.0097 | | 0.0100 |
| 6 | 570 | 00:00:46 | 100.00% | 96.15% | 0.0091 | 0.1386 | 0.0100 |
| 6 | 600 | 00:00:48 | 99.22% | 96.44% | 0.0345 | 0.1270 | 0.0100 |
| 6 | 630 | 00:00:51 | 99.22% | 96.40% | 0.0444 | 0.1348 | 0.0100 |
| 6 | 650 | 00:00:52 | 100.00% | | 0.0076 | | 0.0100 |
| 6 | 660 | 00:00:53 | 100.00% | 96.55% | 0.0141 | 0.1280 | 0.0100 |
| 6 | 666 | 00:00:54 | 100.00% | 96.57% | 0.0118 | 0.1298 | 0.0100 |

Training finished: Max epochs completed.

Fig. 3.10: Details training procedure report of the proposed HDC model

The above figure describes a whole overview of the training procedure. At the last stage of Epoch 1, it can be observed that having 100 iterations elapsing 10 seconds the mini-batch accuracy is 96.88%. At the last stage of the second epoch, the iteration rate is 210 elapsing 18 seconds. In this case, the mini-batch accuracy, mini-batch loss, and validation loss are 96.09%, 95.12%, and 0.1163 respectively. The initial state of Epoch 6 preserves an iteration of 570 and the model needs 46 seconds to reach this stage having a mini-batch accuracy of 100.00%. While validation accuracy, mini-batch loss, and validation loss are 96.15%, 0.0091, and 0.1386 respectively. A scatter graph between two rows of features in the training procedure is expressed in Fig. 3.11.

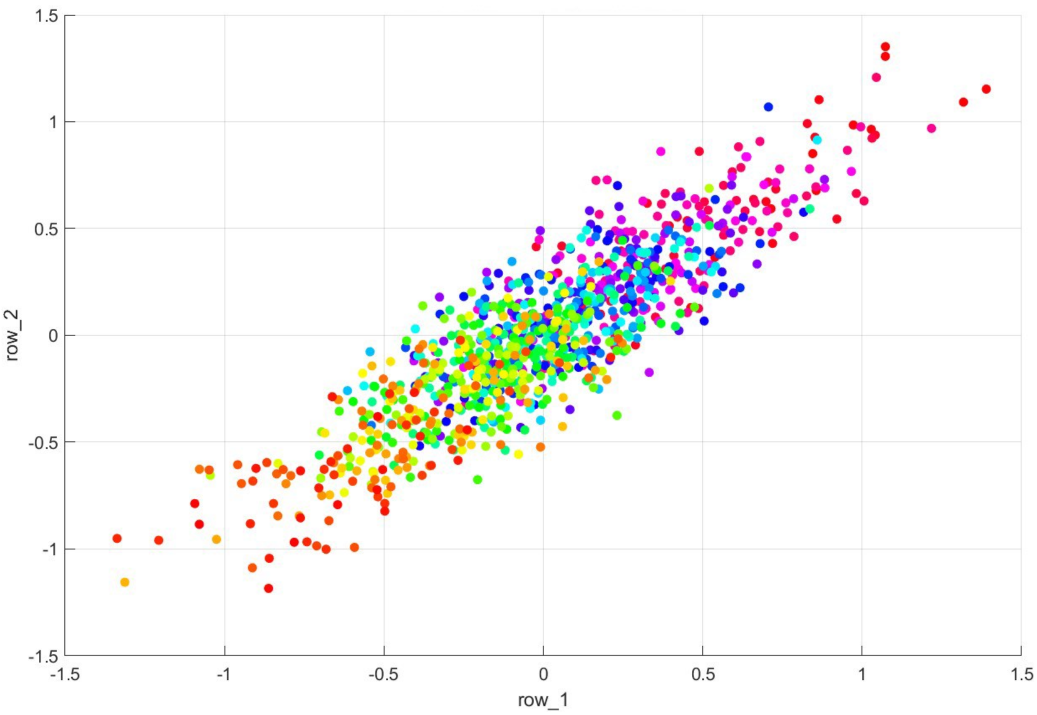


Fig. 3.11: Scatter graph between two rows of features during the training procedure

The graph shows a representative uniform distribution of features at the time of learning of the proposed HDC model between two feature rows.

The confusion matrix for assessment of the projected HDC structure generated in MATLAB is expressed in Fig. 3.12 for better evaluation.

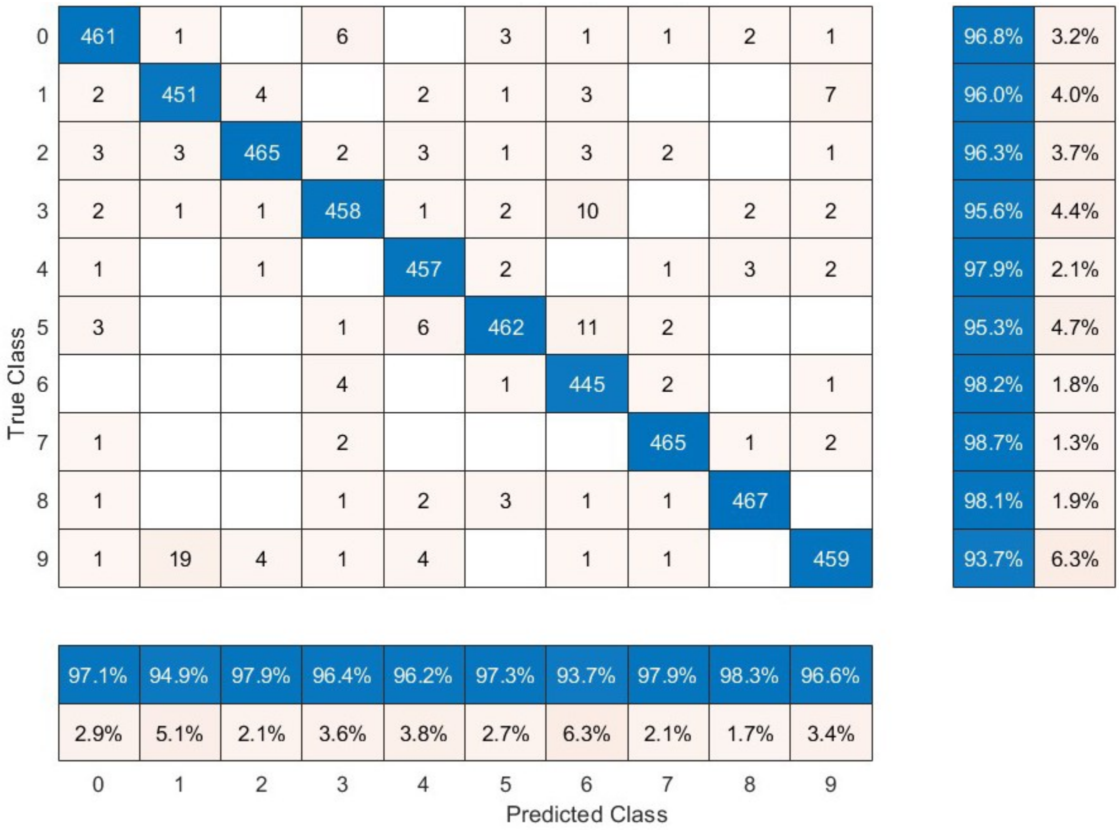


Fig. 3.12: Plotted Confusion Matrix in the perspective of True Class and Predicted Class

Here in the confusion matrix, the 10 handwritten digit classes are indicating appropriately on the side of true class and predicted class. The number of properly and imperfectly classified observations for each predicted class is expressed in the column-normalized column summary as a proportion of the entire number of annotations for that predicted class. The row-normalized row summary displays, as a proportion of the overall quantity of observations for a given true class, the number of acceptably and erroneously classified observations for each true class. The overall training summary of the developed HDC model is expressed in Table 3.5.

Table 3.5: Validation parameter of the developed HDC model

| Parameter | Value |
|---|---|
| Accuracy | 96.72 % |
| Time required | 55 seconds |
| Epoch | 6 of 6 |
| Iteration | 666 of 666 |
| Iterations per epoch | 111 |
| Maximum iterations | 666 |
| Validation Frequency | 30 iterations. |

The HDC model shows an accuracy of 96.72% consuming 55 seconds. Here for better observation of the projected HDC model, significant indicator values such as Precision, Recall, and F1 Score are evaluated in this model for all the 10-digit classes of handwritten Bangla digits, which is shown in Fig. 3.13.

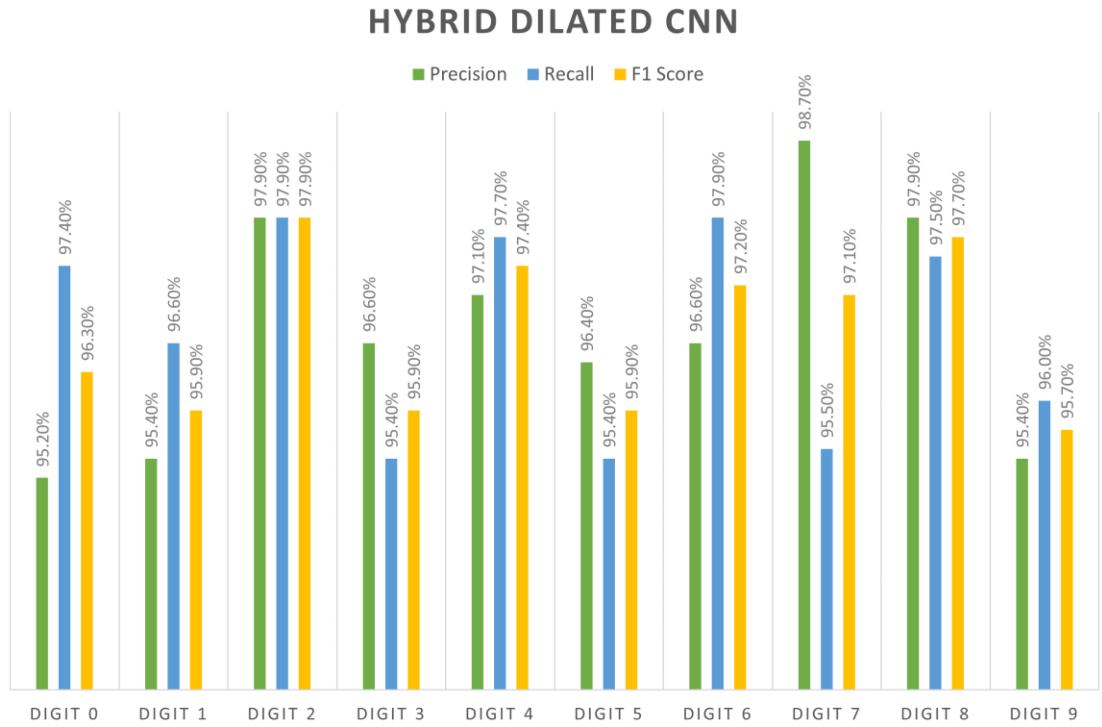


Fig. 3.13: Precision, Recall, and F1 Score evaluation graph for all categories in the HDC model

According to Fig. 3.13, the precision, recall, and f1 score for the handwritten Bangla digit 0 are 95.20%, 97.40%, and 96.03%, respectively. Precision, recall, and f1 score for Bangla digit 1 are 95.40%, 96.60%, and 95.90%, respectively. These values for Bangla digit 2 are all same which is 97.90%. These values for Bangla digit 3 are 96.60%, 95.40%, and 95.90%, respectively. These standards for Bangla digit 4 are 97.10%, 97.70%, and 97.40%, respectively. These standards are 96.40%, 95.40%, and 95.90% in the Bangla digit 5 scenario. For Bangla digit 6, these values are 96.60%, 97.90%, and 97.20% respectively. For Bangla digit 7, these standards are 98.70%, 95.50%, and 97.10% respectively. In the case of Bangla digit 8, these metrics are 97.90%, 97.50%, and 97.70% respectively. In the scenario of Bangla digit 9 these standards are 95.40%, 96.00%, and 95.70% respectively. Out of all the examples the model anticipated would be of the positive class, precision is the proportion of tasters that belong to the positive class. Out of all the samples that belong to the positive class, recall is the proportion of samples that were precisely prophesied to do so. The F1 score stands for the positive class resulting from the calculation of recall and precision. By carefully examining all the results, it is clear that the Hybrid Dilated CNN (HDC) is quite effective at identifying handwritten Bangla digits from 0 to 9.

## 3.9 PARAMETER EVALUATION OF THE PROPOSED CNN, DILATED CNN, AND HYBRID DILATED CNN (HDC) MODEL

To observe the effect of change in dilation rate in the proposed models, all the parameters of the Convolution Layers are kept the same. The calculation of maintained parameters in the Convolution Layers of all the proposed models is shown in this section. Here the total amount of parameters in a Convolution Layer can be identified using the values of weights and biases utilizing specific mathematical equations [93].

**Now let's define,**

$B_{CL} = Number\ of\ Biases\ of\ the\ Convolution\ Layer$

$W_{CL} = Number\ of\ Weights\ of\ the\ Convolution\ Layer$

$P_{CL} = Number\ of\ Parameters\ of\ the\ Convolution\ Layer$

$S_K = Size\ (width)\ of\ Kernels\ used\ in\ the\ Convolution\ Layer$

$N_K = Number\ of\ Kernels\ used\ in\ the\ Convolution\ Layer$

$\&\ N_C = Number\ of\ Channels\ of\ the\ input\ image$

$$Here, W_{CL} = S_K{}^2 \times N_C \times N_K \quad (3.3)$$

$$B_{CL} = N_K \quad (3.4)$$

$$P_{CL} = W_{CL} + B_{CL} \quad (3.5)$$

**<u>In case of 1st Convolution Layer of the proposed CNN, Dilated CNN and Hybrid Dilated CNN models:</u>**

$Here,$

$Size\ (width)\ of\ Kernels\ used\ in\ the\ Convolution\ Layer, S_K = 3$

$Number\ of\ Kernels\ used\ in\ the\ Convolution\ Layer, N_K = 8$

$\&\ Number\ of\ Channels\ of\ the\ input\ image, N_C = 3$

$The\ number\ of\ Weights\ of\ the\ Convolution\ Layer\ is,$

$$W_{CL} = S_K{}^2 \times N_C \times N_K$$
$$= 3^2 \times 3 \times 8$$
$$= 216$$

$The\ number\ of\ Biases\ of\ the\ Convolution\ Layer\ is,$

$$B_{CL} = N_K$$
$$= 8$$

$Therefore,$

$Number\ of\ Parameters\ of\ the\ Convolution\ Layer\ is,$

$$P_{CL} = W_{CL} + B_{CL}$$

$= 216 + 8$

$= 224$

**<u>In case of 2nd Convolution Layer of the proposed CNN, Dilated CNN and Hybrid Dilated CNN models:</u>**

$Here,$

$Size\ (width)\ of\ Kernels\ used\ in\ the\ Convolution\ Layer, S_K = 3$

$Number\ of\ Kernels\ used\ in\ the\ Convolution\ Layer, N_K = 16$

$\&\ Number\ of\ Channels\ of\ the\ input\ image, N_C = 8$

$The\ number\ of\ Weights\ of\ the\ Convolution\ Layer\ is,$

$W_{CL} = {S_K}^2 \times N_C \times N_K$

$= 3^2 \times 8 \times 16$

$= 1152$

$The\ number\ of\ Biases\ of\ the\ Convolution\ Layer\ is,$

$B_{CL} = N_K$

$= 16$

$Therefore,$

$Number\ of\ Parameters\ of\ the\ Convolution\ Layer\ is,$

$P_{CL} = W_{CL} + B_{CL}$

$= 1152 + 16$

$= 1168$

**<u>In case of 3rd Convolution Layer of the proposed CNN, Dilated CNN and Hybrid Dilated CNN models:</u>**

$Here,$

$Size\ (width)\ of\ Kernels\ used\ in\ the\ Convolution\ Layer, S_K = 3$

$Number\ of\ Kernels\ used\ in\ the\ Convolution\ Layer, N_K = 32$

$\&\ Number\ of\ Channels\ of\ the\ input\ image, N_C = 16$

$The\ number\ of\ Weights\ of\ the\ Convolution\ Layer\ is,$

$$W_{CL} = {S_K}^2 \times N_C \times N_K$$

$$= 3^2 \times 16 \times 32$$

$$= 4608$$

$The\ number\ of\ Biases\ of\ the\ Convolution\ Layer\ is,$

$$B_{CL} = N_K$$

$$= 32$$

$Therefore,$

$Number\ of\ Parameters\ of\ the\ Convolution\ Layer\ is,$

$$P_{CL} = W_{CL} + B_{CL}$$

$$= 4608 + 32$$

$$= 4640$$

**<u>In case of 4th Convolution Layer of the proposed CNN, Dilated CNN and Hybrid Dilated CNN models:</u>**

$Here,$

$Size\ (width)\ of\ Kernels\ used\ in\ the\ Convolution\ Layer, S_K = 3$

$Number\ of\ Kernels\ used\ in\ the\ Convolution\ Layer, N_K = 64$

$\&\ Number\ of\ Channels\ of\ the\ input\ image, N_C = 32$

$The\ number\ of\ Weights\ of\ the\ Convolution\ Layer\ is,$

$$W_{CL} = {S_K}^2 \times N_C \times N_K$$

$$= 3^2 \times 32 \times 64$$

$$= 18432$$

$The\ number\ of\ Biases\ of\ the\ Convolution\ Layer\ is,$

$B_{CL} = N_K$

$= 64$

$Therefore,$

$Number\ of\ Parameters\ of\ the\ Convolution\ Layer\ is,$

$P_{CL} = W_{CL} + B_{CL}$

$= 18432 + 64$

$= 18496$

**In case of 5th Convolution Layer of the proposed CNN, Dilated CNN and Hybrid Dilated CNN models:**

$Here,$

$Size\ (width)\ of\ Kernels\ used\ in\ the\ Convolution\ Layer, S_K = 3$

$Number\ of\ Kernels\ used\ in\ the\ Convolution\ Layer, N_K = 64$

$\&\ Number\ of\ Channels\ of\ the\ input\ image, N_C = 32$

$The\ number\ of\ Weights\ of\ the\ Convolution\ Layer\ is,$

$W_{CL} = {S_K}^2 \times N_C \times N_K$

$= 3^2 \times 64 \times 128$

$= 73728$

$The\ number\ of\ Biases\ of\ the\ Convolution\ Layer\ is,$

$B_{CL} = N_K$

$= 128$

$Therefore,$

$Number\ of\ Parameters\ of\ the\ Convolution\ Layer\ is,$

$P_{CL} = W_{CL} + B_{CL}$

$= 73728 + 128$

$= 73856$

**<u>In case of 6th Convolution Layer of the proposed CNN, Dilated CNN and Hybrid Dilated CNN models:</u>**

$Here,$

$Size\ (width)\ of\ Kernels\ used\ in\ the\ Convolution\ Layer, S_K = 3$

$Number\ of\ Kernels\ used\ in\ the\ Convolution\ Layer, N_K = 64$

$\&\ Number\ of\ Channels\ of\ the\ input\ image, N_C = 32$

$The\ number\ of\ Weights\ of\ the\ Convolution\ Layer\ is,$

$W_{CL} = {S_K}^2 \times N_C \times N_K$

$= 3^2 \times 128 \times 256$

$= 294912$

$The\ number\ of\ Biases\ of\ the\ Convolution\ Layer\ is,$

$B_{CL} = N_K$

$= 256$

$Therefore,$

$Number\ of\ Parameters\ of\ the\ Convolution\ Layer\ is,$

$P_{CL} = W_{CL} + B_{CL}$

$= 294912 + 256$

$= 295168$

From the above mathematical calculation, it can be observed that by keeping the learnable parameters same and utilizing the mechanism of the convolution layer's dilation effect, significant improvement can be achieved. Here, it is evident that all of the learnable parameters in the corresponding convolution layers of the CNN, Dilated CNN, and Hybrid Dilated CNN (HDC) models are identical. The dilation rate of the convolution layers is designed differently. This adjustment has an impact on how feature maps capture pixels. Based on the results discussed in earlier sections, it is clear that a well-designed dilation rate can significantly improve system performance without interfering with other parameters of the traditional models.

### 3.10 HYBRID MACHINE LEARNING CLASSIFICATION

To develop a better Bangla handwritten digit recognition, this research also emphasizes on machine learning algorithms. Two prominent machine learning systems Support Vector Machine (SVM) [94] and K-Nearest Neighbors (KNN) [95] are instigated with the projected Hybrid Dilated CNN (HDC) model. The extracted features from the HDC structure are used to train the SVM and CNN for classification tasks as these output features can be used as features for any classifier for classification tasks [30].

Firstly, the HDC model is trained by the handwritten digit dataset for learning. The Fully Connected Layer needs to be used as the output layer during the training of a convolutional neural network because of the procedure of learning. Then the learned model produces a features vector. These output features can be used as features for any classifier for classification tasks [30].

Secondly, to implement the hybrid HDC-SVM/KNN model, the output layer of the developed HDC model is swapped by SVM and KNN respectively to develop two different hybrid deep learning models. Where HDC (Hybrid Dilated CNN) is utilized for feature extraction and SVM/KNN is employed for classification. Fusion of the HDC and SVM/KNN is mentioned in this effort as a 'hybrid deep learning model'. Here SVM and KNN are trained with the feature vector formed by the HDC model. The parameters of the training procedure remain the same with the HDC model for pinpoint evaluation.

A basic block diagram of the entire technique of the hybrid deep learning model is expressed in Fig. 3.14.

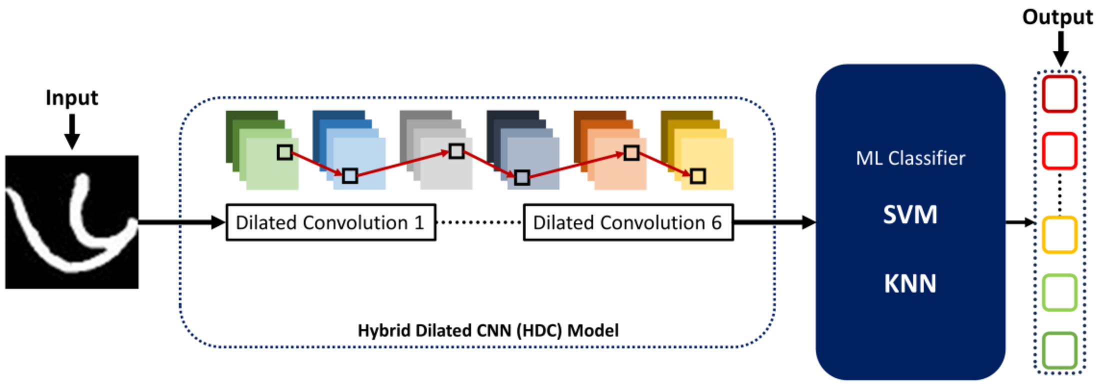


Fig. 3.14: Block illustration of the whole procedure of the hybrid machine learning classification

### 3.10.1 Design of HDC – SVM Model

HDC – SVM model is implemented by utilizing the output feature vector of the HDC structure as input for the SVM classifier.

#### *3.10.1.1 Training and Validation of HDC – SVM Model*

Training parameters of the proposed SVM classifier are expressed in Table 3.6.

Table 3.6: Training specifications of the proposed SVM Classifier

| Parameters of SVM | Value |
|---|---|
| CodingName | 'onevsone' |
| RowsUsed | [ ] |
| ScoreTransform | 'none' |
| BinaryLoss | 'hinge' |
| BinaryLearners | 1x1 |
| VerbosityLevel | 0 |
| NumConcurrent | [ ] |
| FitPoosterior | 0 |

To assess the proposed HDC – SVM structure the confusion matrix is revealed in Fig. 3.15.

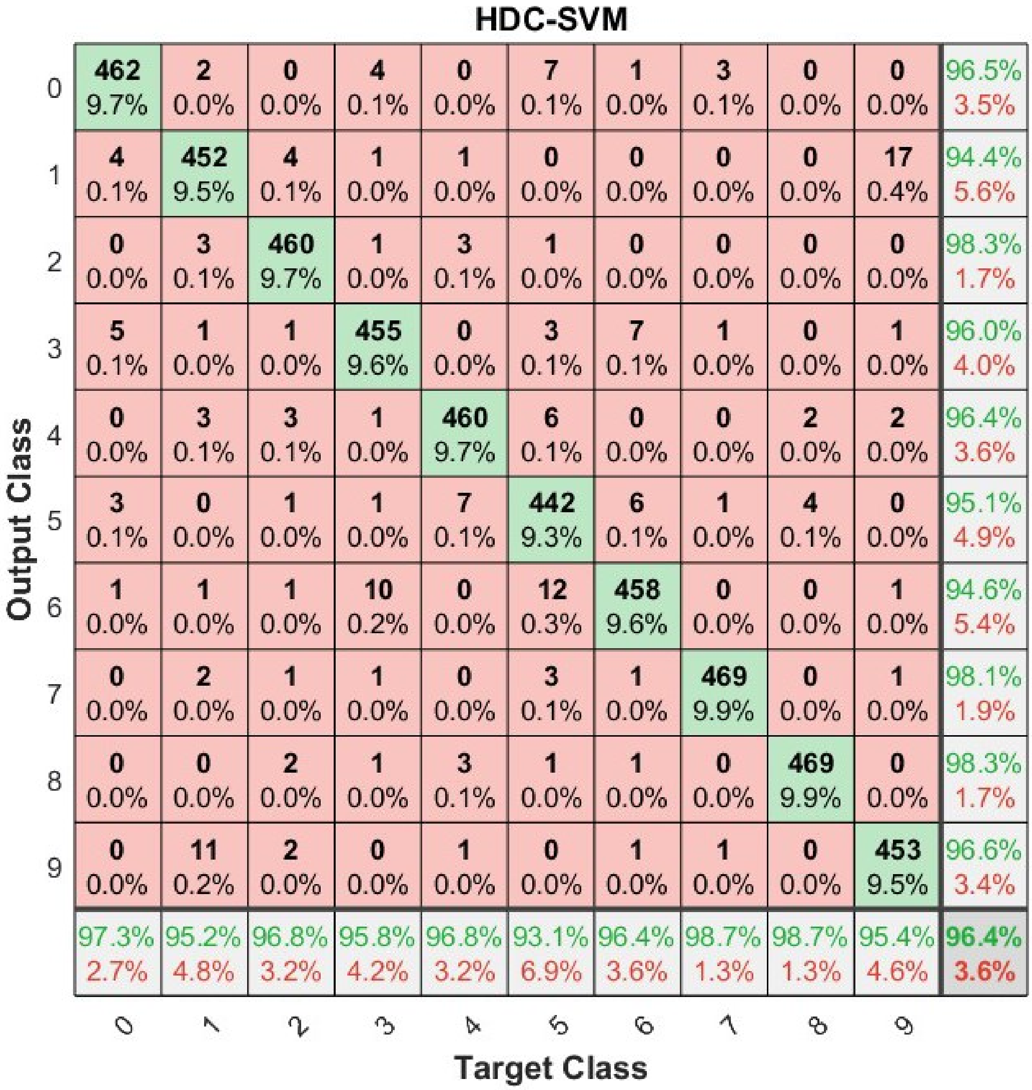


Fig. 3.15: Confusion Matrix of the HDC – SVM structure

In Figure 3.15, the rows denote the expected class and the columns the actual class. With proper categorization, the diagonal cells resemble to observations. The observations that were incorrectly classified are linked to the off-diagonal cells. Both the total quantity of observations and the proportion of all observations are shown in each cell. The cell in the lower right corner of the plot expresses the inclusive exactness [96]-[97].

Here, the HDC – SVM model showed an accurateness of 96.4% which is slightly less than the HDC model.

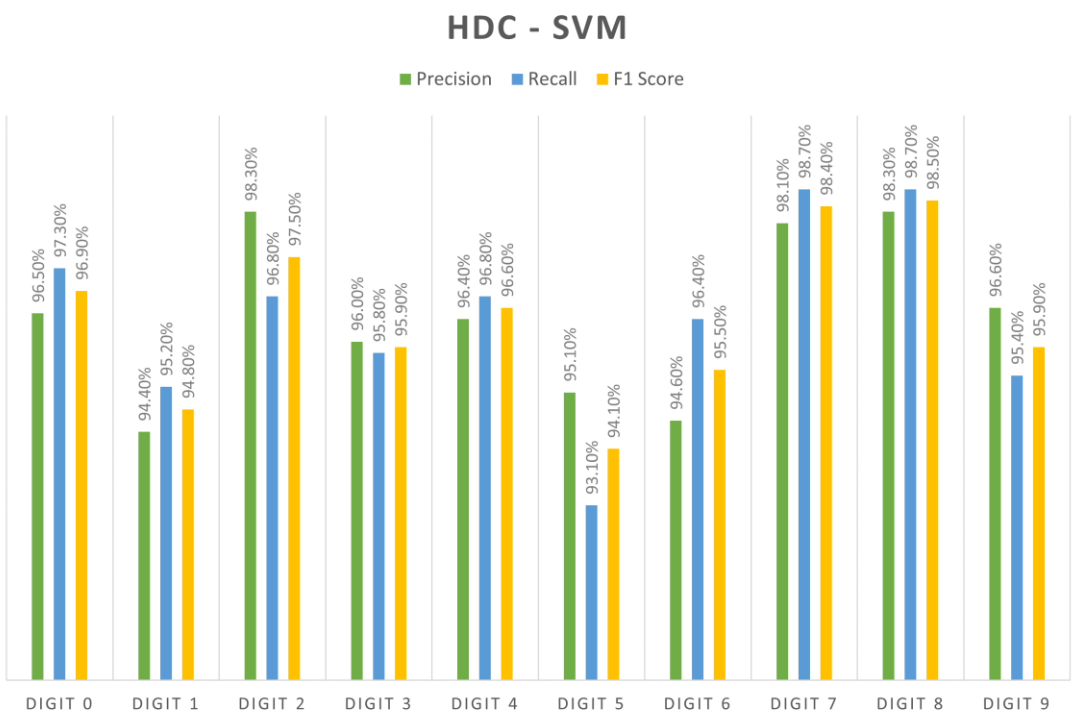


Fig. 3.16: Precision, Recall, and F1 Score evaluation graph for all categories in HDC – SVM model

As the accuracy rate decreases, individual class recognition scenario is observed to identify the more complex digit to recognize in the HDC – SVM model. Precision, Recall, and F1 Score are calculated and expressed in Fig. 3.16 for all 10 digits. From Fig. 3.16, it can be observed that digit 1, digit 5, and digit 6 show complexity in terms of recognition.

### 3.10.2 Design of HDC – KNN Model

HDC – KNN model is implemented by using the output feature vector of the HDC model as input for the KNN classifier.

The KNN algorithm chooses K neighbors at the beginning. The program then determines the Manhattan distance or Euclidean distance between K neighbors. The classifier picks the K nearest neighbors based on the determined Manhattan or Euclidean distance after performing mathematical operations

[98]-[101]. The classifier then counts how many new data types there are among those k nearest neighbors for each data type.

When there are the most neighbors for the new data type, the KNN classifier assigns it to the matching existing data type. This is how the suggested HDC – KNN model functions in the work.

The process of calculation of Euclidean distance between data types [100]-[101] is expressed in Fig. 3.17.

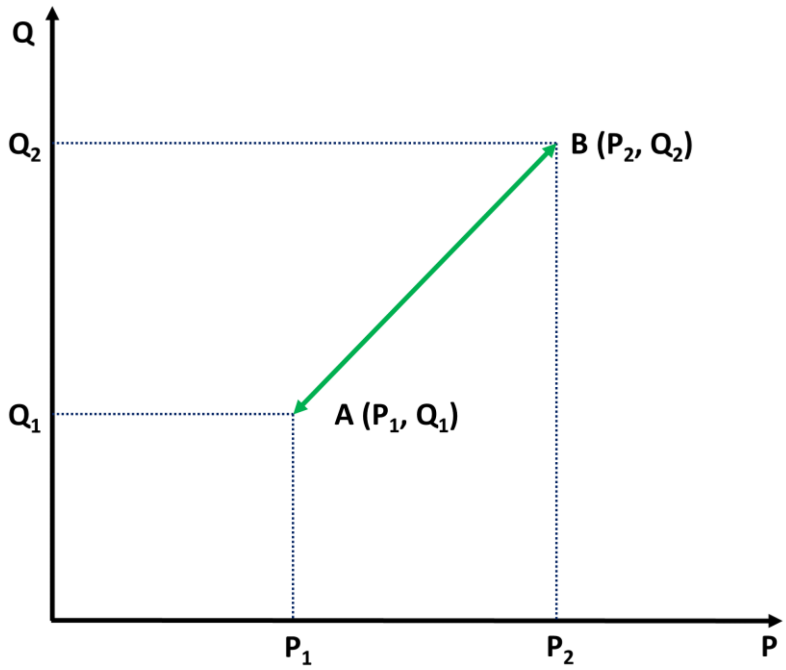


Fig. 3.17: Calculation course of Euclidean Distance in KNN algorithm for two different data types

Here by observing Fig. 3.17 the Euclidean Distance between A and B data types can be expressed as

$$D_p = \sqrt{(P_2 - P_1)^2 + (Q_2 - Q_1)^2} \qquad (3.6)$$

Taking all necessary measures hybrid HDC - KNN model is formed and simulated in MATLAB software environment.

Training parameters of the proposed KNN architecture are expressed in Table 3.7.

Table 3.7: Training specifications of the projected KNN model

| **Parameters of KNN** | **Value** |
|---|---|
| NumNeighbors | 1 |
| Distance | ‘euclidean’ |
| DistParameter | [ ] |
| IncludeTies | 0 |
| DistanceWeight | ‘equal’ |
| BreakTies | ‘smallest’ |
| NSMethod | ‘eshaustive’ |
| Mu | [ ] |
| Sigma | [ ] |
| ResponseName | ‘Y’ |
| ScoreTransform | ‘none’ |
| Exponent | [ ] |
| BucketSize | [ ] |
| StandardizeData | 0 |

#### *3.10.2.1 Training and Validation of HDC – KNN Model on BanglaLekha-Isolated Dataset*

To evaluate the proposed HDC – KNN model for handwritten Bangla digit the confusion matrix is revealed in Fig. 3.18.

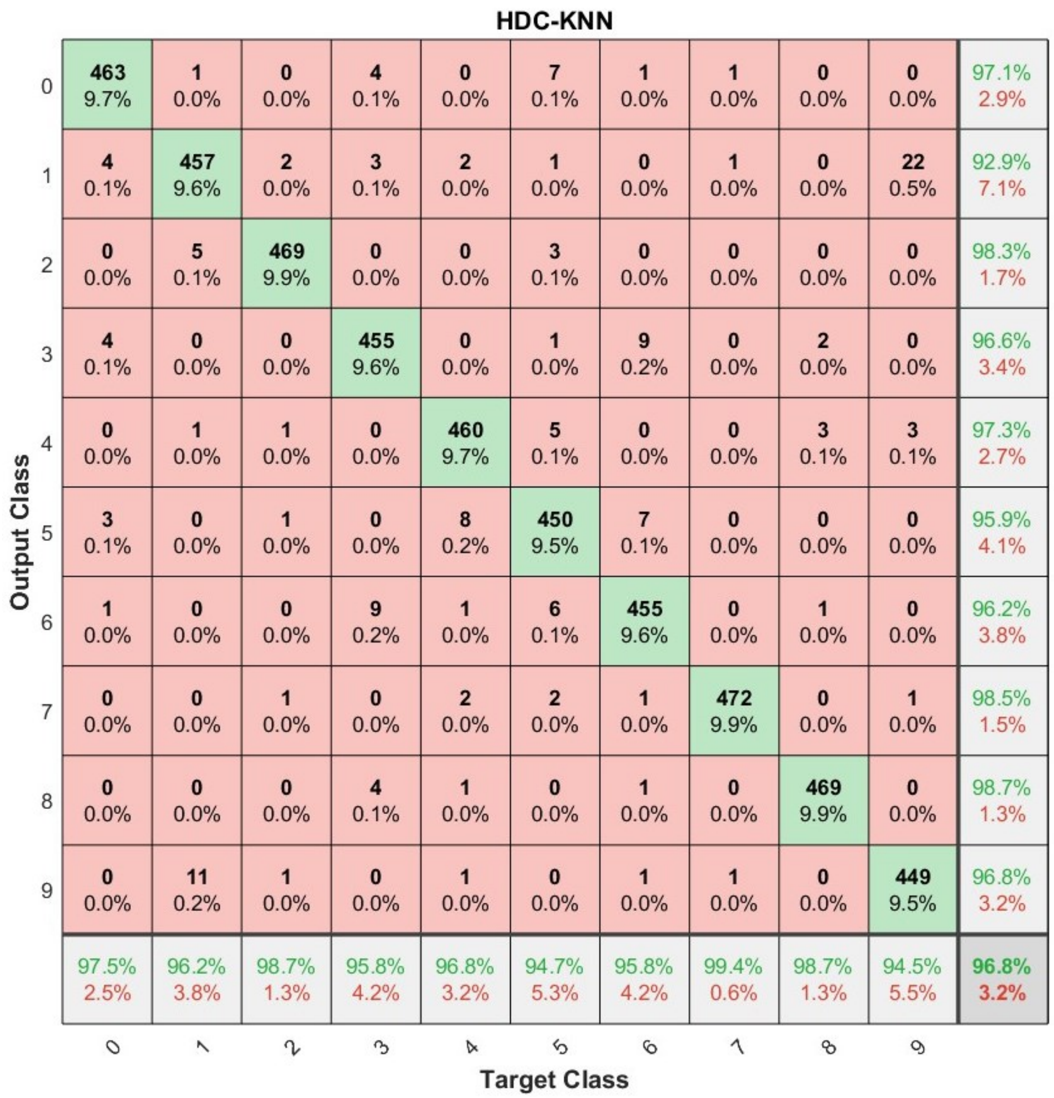


Fig. 3.18: Confusion Matrix of the HDC – KNN structure (Bangla Digit)

In Fig. 3.18, the anticipated class is epitomized by rows, and the actual class is embodied by columns. The diagonal cells match the appropriately classified observations. The observations that were incorrectly classified are linked to the off-diagonal cells. Both the total quantity of observations and the proportion of all observations are shown in each cell. The cell in the lower right corner of the plot expresses the overall accurateness [96]-[97].

Here, the HDC – KNN model shows an accuracy of 96.8%. The accuracy rate is noteworthy and surpasses the other models proposed in this paper.

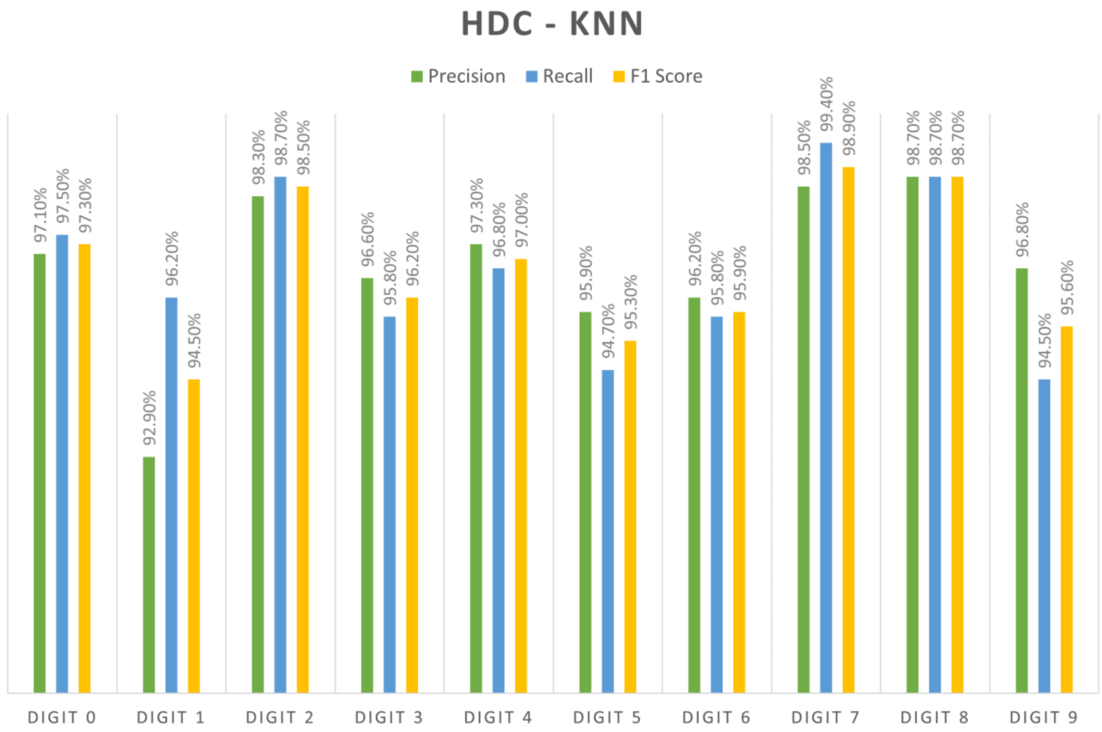


Fig. 3.19: Precision, Recall, and F1 Score evaluation graph for all categories in HDC – KNN model (Bangla Digit)

As the HDC – KNN model performs very well, this research did a deep analysis for all the 10-digit classes. For proper evaluation and analysis, this research focuses on several parameters such as Precision, Recall, and F1 Score for the HDC – KNN model which is expressed in Fig. 3.19. Precision is demarcated here as the proportion of samples out of all samples that the model predicted to be in the positive class that really belonged to it. Precision can be defined by the following equation [102]-[103].

$$Precision = \frac{TP}{TP + FP} \qquad (3.7)$$

True Positive (TP) in this context refers to a sample that was accurately acknowledged as being in the positive class. False Positive, or FP, refers to a sample that should be classed as negative but is instead mistakenly placed in the positive type.

The quantity of samples accurately identified as belonging to the positive class out of all the tasters that really do so is represented by the value of Recall in this instance. Recall can be defined by the following equation [102], [104].

$$Recall = \frac{TP}{TP + FN} \qquad (3.8)$$

Here, TP stands for true positive, which refers to a sample that is accurately identified as being in the positive class, and FN stands for false negative, which refers to a taster that is appropriately classified as being in the positive class but is actually in the wrong class.

By using the value of Precision and Recall in the case of every digit category, the F1 Score derives as a consonant mean value. The perfection of the HDC – KNN model is easily identifiable by observing these values for each class. For the class digit 0, the value of precision, recall, and f1 score are 97.10%, 97.50%, and 97.30% respectively. In the case of digit 1, these values are 92.90%, 95.60%, and 94.50% respectively. For digit 2, these standards are 98.30%, 98.70%, and 98.50% respectively. In the scenario of digit 3, these metrics are 96.60%, 95.80%, and 96.20% respectively. For digit 4, these values are 97.30%, 95.80% and 97.00% respectively. For digit 5, these values are 95.90%, 94.70%, and 95.30% respectively. For digit 6, these values are 96.20%, 95.80%, and 95.90% respectively. For digit 7, these values are 98.50%, 99.40%, and 98.90% respectively. In the case of digit 8, these standards are 98.70%, 98.70%, and 98.70% respectively. In the scenario of digit 9, these metrics are 96.80%, 94.50%, and 95.60% respectively. By observing these values for each class, it can easily be understandable that the projected HDC – KNN structure is exceedingly efficient in recognizing handwritten Bangla digits.

#### *3.10.2.2 Training and Validation of HDC – KNN Model on MNIST Dataset*

To evaluate the proposed HDC – KNN structure for handwritten English digit appreciation the confusion matrix is revealed in Fig. 3.20.

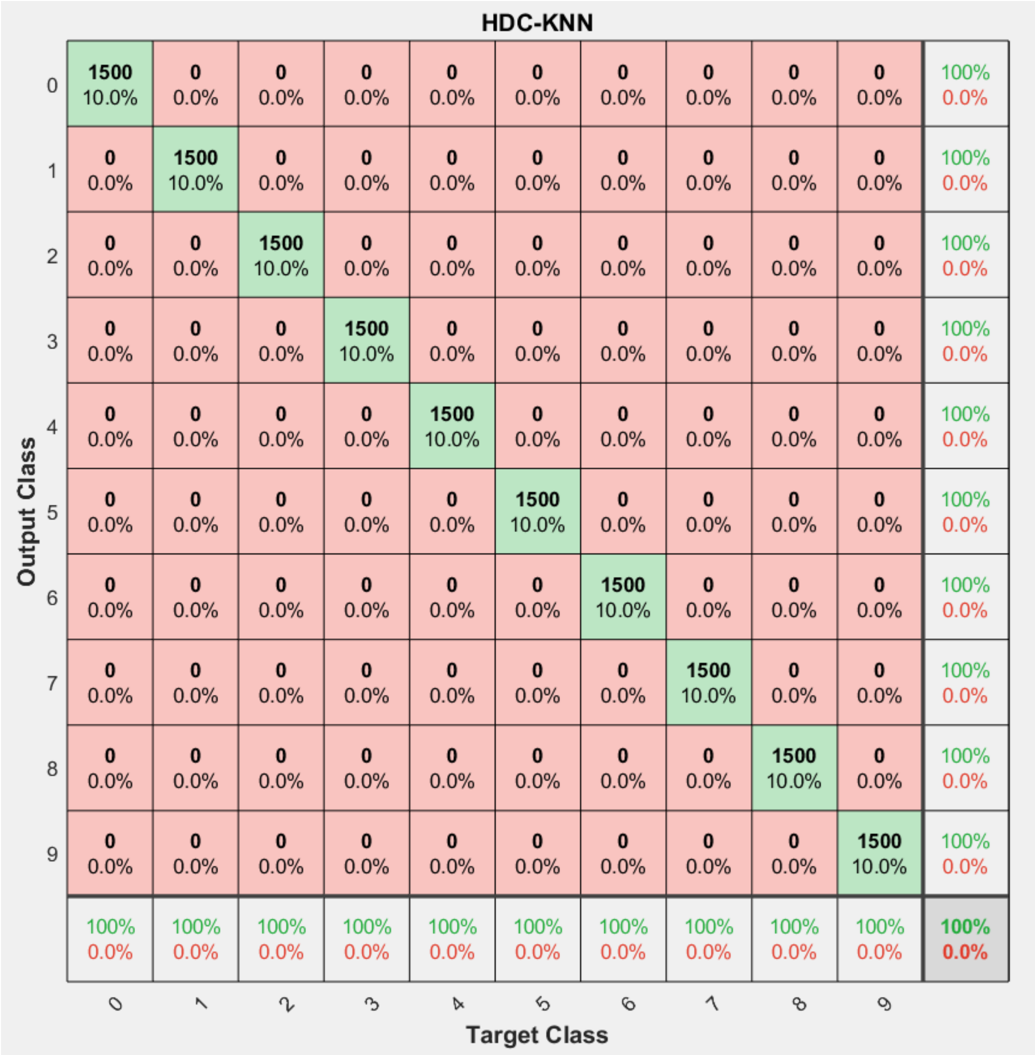


Fig. 3.20: Confusion Matrix of the HDC – KNN model (English Digit)

In Fig. 3.20, the anticipated class is epitomized by rows, while the concrete class is embodied by columns. The diagonal cells match the appropriately classified observations. The observations that were incorrectly classified are linked to the off-diagonal cells. Both the total quantity of observations and the proportion of all observations are shown in each cell. The cell in the lower right corner of the plot expresses the overall accurateness [96]-[97].

Here, the HDC-KNN model shows an accuracy of 100%. The accuracy rate is noteworthy and surpasses the other models proposed in this paper.

However, the number of datasets images utilized in this scenario is larger than the Bangla digit dataset samples. This increased size of dataset samples also plays a vital role for this noteworthy accuracy.

To evaluate the proposed HDC – KNN model the training progress graph is shown in Fig. 3.21. From the figure, it can be distinguished that training finished successfully elapsing 2 minutes 52 seconds.

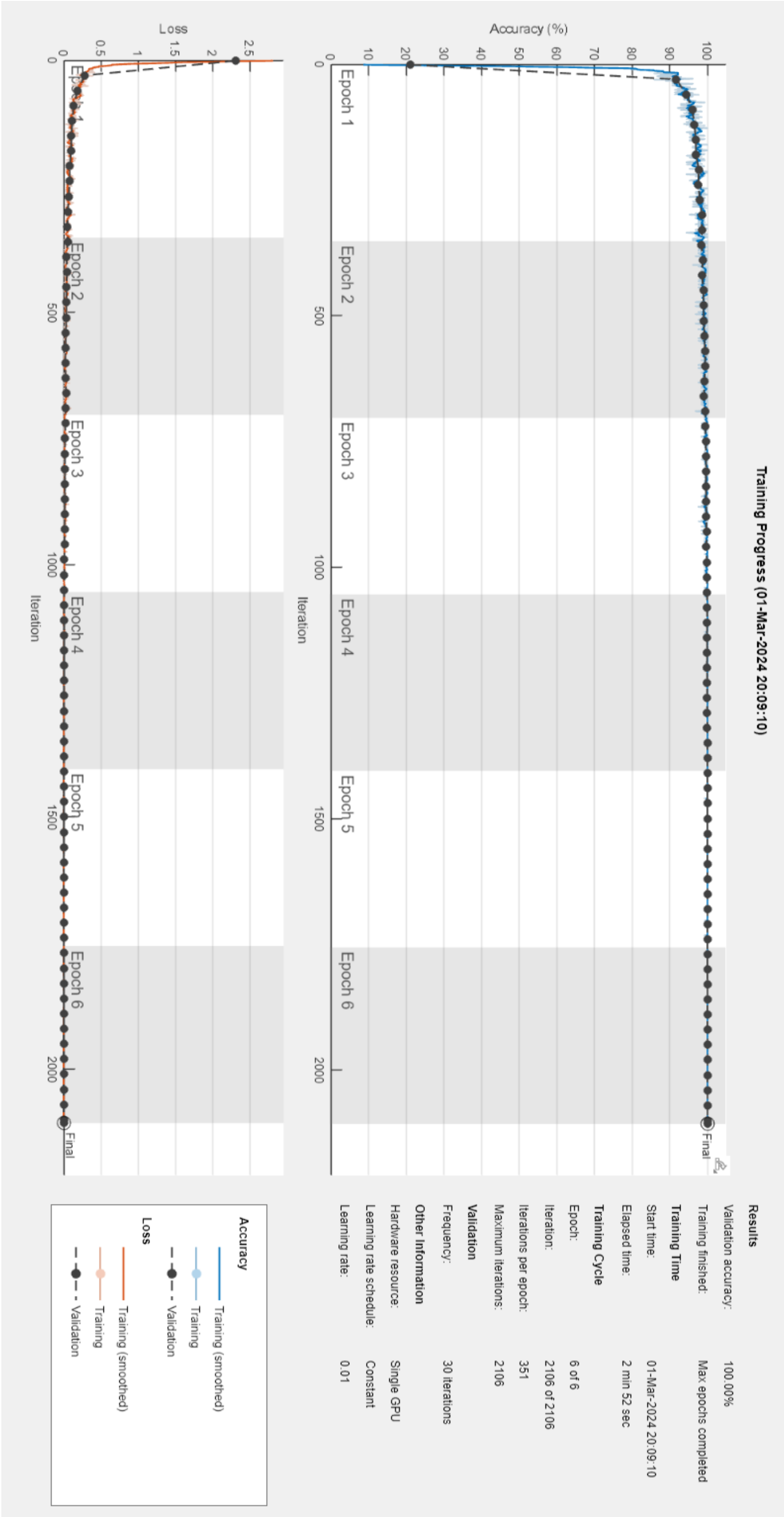


Fig. 3.21: Training progress graph of the proposed HDC – KNN model for English digit

Since the HDC – KNN model works so effectively, all ten-digit classes are thoroughly analyzed in this research. This study focuses on several system of measurement, including Precision, Recall, and F1 Score for the HDC – KNN model, which is represented in Fig. 3.22, to properly evaluate and analyze the data.

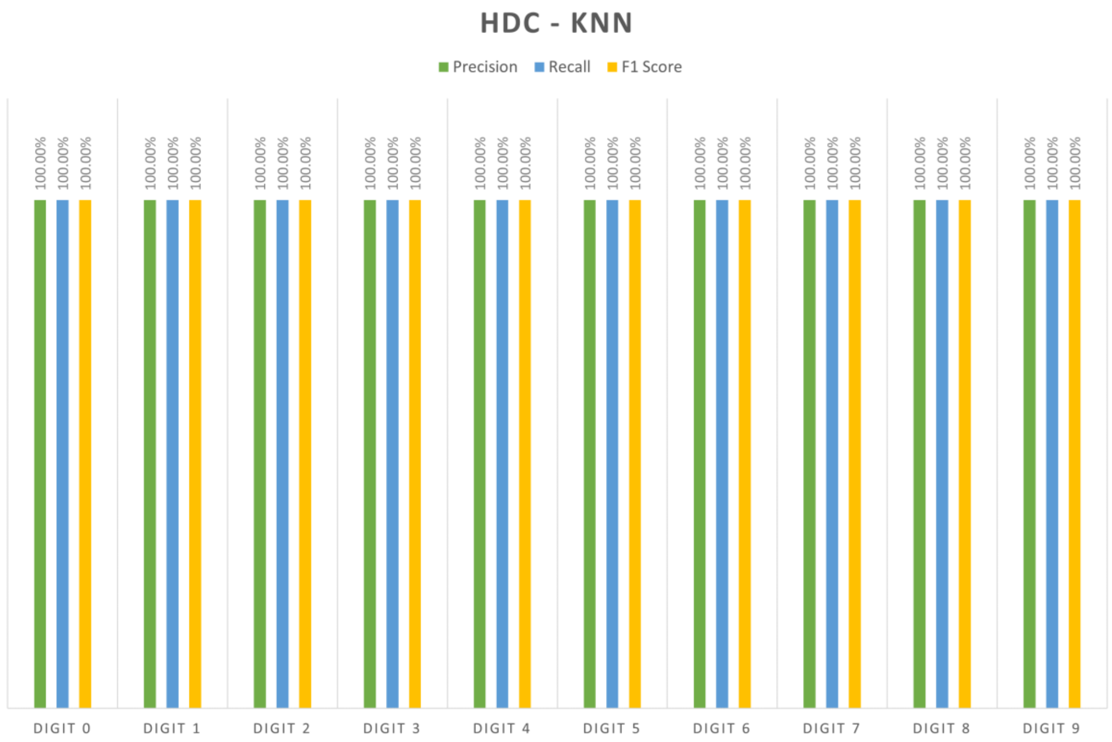


Fig. 3.22: Precision, Recall, and F1 Score evaluation graph for all categories in HDC – KNN model (English Digit)

By using the value of Precision and Recall in the case of every digit category, the F1 Score derives as a consonant mean value. The perfection of the HDC – KNN model is easily identifiable by observing these values for each class. From Fig. 3.22 it can be observed that, there are ten English digit classes- which are digit 0, digit 1, digit 2, digit 3, digit 4, digit 5, digit 6, digit 7, digit 8, and, digit 9. The values of precision, recall, and f1 score for all these above-mentioned digit classes in case of HDC – KNN model are 100%, 100%, and, 100% respectively. By observing these values for each class, it can easily be understandable that the proposed HDC – KNN model is highly efficient in recognizing handwritten English digits.

# Chapter 4: RESULTS AND ANALYSIS

*In this chapter, the suggested models' results are analyzed, and their comparison with existing models based on the BanglaLekha-Isloated Dataset, MNIST Dataset and other Datasets is assessed.*

## 4.1 RESULT ANALYSIS AND DISCUSSION

All the proposed models in this research are mentioned in Table 4.1 along with computation time and accuracy.

Table 4.1: Comparison among developed models

| **Model Name** | **Dataset** | **Computation Time (Seconds)** | **Accuracy (%)** |
|---|---|---|---|
| CNN | BanglaLekha-Isolated | 63 | 96.48 |
| Dilated CNN | BanglaLekha-Isolated | 52 | 95.85 |
| Hybrid Dilated CNN (HDC) | BanglaLekha-Isolated | 55 | 96.72 |
| HDC – SVM | BanglaLekha-Isolated | 55 | 96.40 |
| HDC – KNN (Bangla Digit) | BanglaLekha-Isolated | 55 | 96.80 |
| HDC – KNN (English Digit) | MNIST | 127 | 100 |

From Table 4.1, it can be distinguished that the Convolutional Neural Network (CNN) shows an accuracy of 96.48%. To perform the whole task the CNN model consumes 63 seconds. As the CNN model uses enormous computing resources and also has a time-consuming problem when layer increases, to solve the issue a Dilated CNN model is proposed. The Dilated CNN structure shows an accuracy of 95.85% and consumes 52 seconds for the whole process. Though the Dilated CNN structure beat the CNN structure in terms of computation time but arises a few new problems. It shows 0.63% less accuracy. This happens as the Dilated CNN model inserted holes in the feature map to cover a larger area to reduce computation time. This also turns into capturing less perfect dataset features as several pixels are omitted to make a feature map.

So, to solve the all difficulties that ascend in the CNN and the Dilated CNN model a unique Hybrid Dilated CNN (HDC) model is expressed in this research. Different dilation rate applied in the convolutional layers in the HDC model makes it capable of reducing computation time and covering more features than previous models. The HDC model takes 3 seconds more computation time than the Dilated CNN model and 8 seconds less time than the CNN model. It shows 96.72% accuracy and surpasses both the CNN and the Dilated CNN model.

However, to develop a state-of-the-art handwritten Bangla digit appreciation system, this research emphasizes on machine learning algorithms for classification tasks. Here, the HDC model is trained firstly with the handwritten Bangla digit dataset. Then the produced features vector from the learned model is utilized as input for machine learning algorithms for classification tasks.

Firstly, a Support Vector Machine (SVM) is utilized for classification tasks. Because, Empirical Risk Minimization, the foundation of MLP, seeks to reduce training set errors. As a result, when the back-propagation process identifies the first separating hyperplane, the training procedure is terminated.

On the other hand, an SVM classifier uses the Structural Risk Minimization principle to try and reduce generalization mistakes. The hyperplane with the greatest margin between two categories of training samples is what the algorithm seeks to identify. Therefore, after substituting the

MLP output components in the CNN, the SVM's generalization capability is increased to advance the hybrid model's classification accurateness and shows a performance improvement when SVM is used along with CNN [1, 30].

So, this research tries to use SVM along with the HDC model as the basic working principle is related to CNN. However, from Table 4.1 it can be distinguished that the HDC – SVM model shows a 96.40% accuracy which is lesser than the HDC model. The reason is that different dilation rates of the HDC model and parameter optimization can cover the above-mentioned limitation.

Then, K-Nearest Neighbors (KNN) is implemented in this work along with the HDC model for the classification task. The KNN’s classification task is based on Euclidean distance measurement among features that work differently than SVM. Here, HDC – KNN models show an accuracy of 96.80% for handwritten Bangla digit recognition which outperforms all other models proposed in this work.

Observing the above parameters MNIST Dataset is utilized in the HDC – KNN model for handwritten English digit recognition and it shows an accuracy of 100% consuming 127 seconds. The scenario has a slightly long processing time because of the large amount of dataset images used.

For more deeper evaluation of the proposed models, a comparison bar graph and table are expressed in Fig. 4.1.

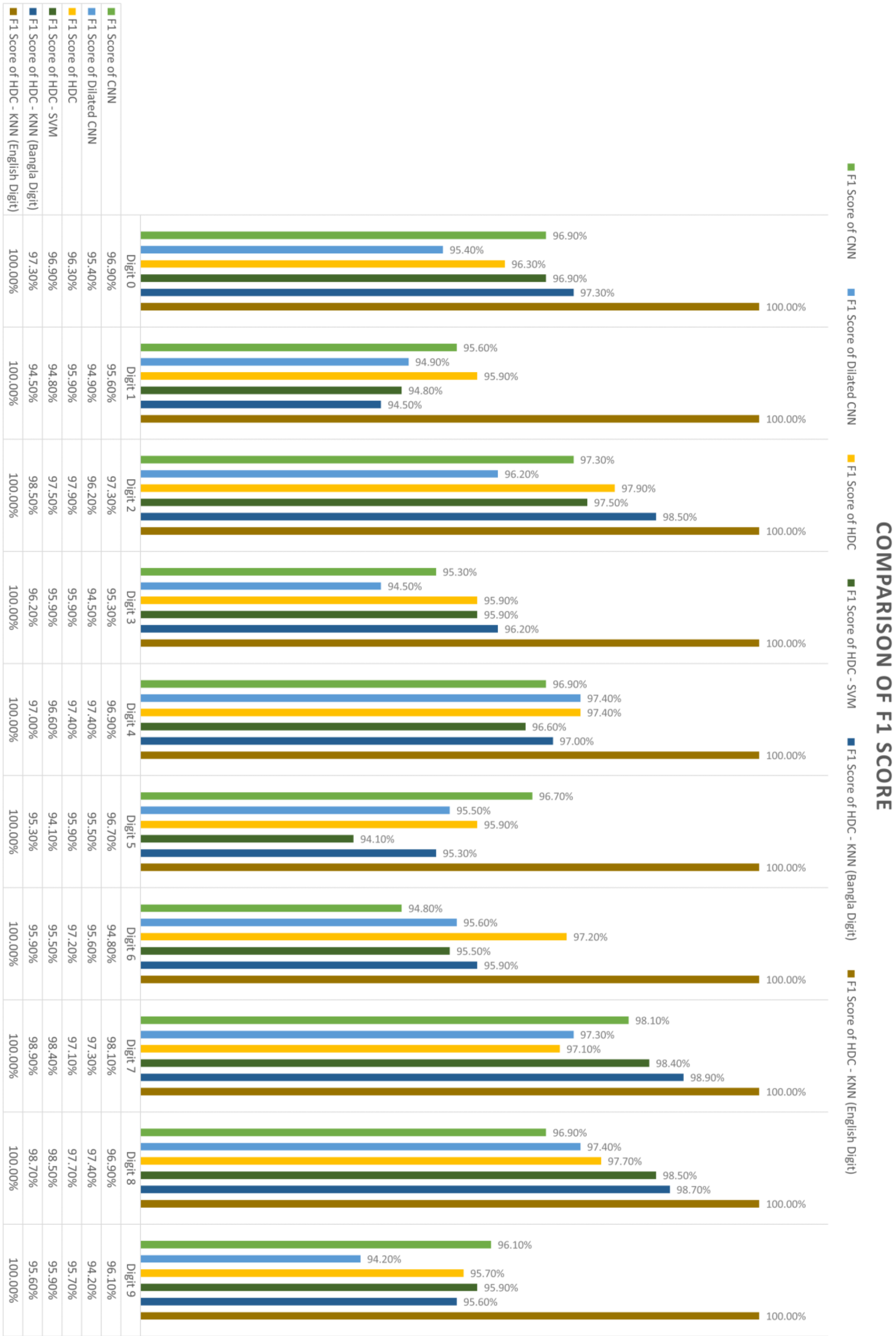


Fig. 4.1: F1 Score evaluation for all the developed models in every handwritten digit class

The F1 Score epitomizes the consonant mean of the Precision and Recall scores accomplished for the positive class in this instance. The value of the F1 Score can be expressed by the following equation [105]-[106].

$$F1\ Score = \frac{2\ \times Precision\ \times\ Recall}{Precision + Recall} \qquad (4.1)$$

Out of all the tasters, the model anticipated would be of the positive class, precision in this case refers to the number of tasters that really belong to the positive class. Out of all the tasters that actually belong to the positive class, recall is the proportion of tasters that were precisely prophesied to do so.

From the Fig. 4.1 it can be distinguished that, the F1 Score of HDC – KNN structure for all the 10 handwritten Bangla digits from digit 0 to digit 9 are 97.30%, 94.50%, 98.50%, 96.20%, 97.00%, 95.30%, 95.90%, 98.90%, 98.70% & 95.60%. These values indicate the perfection of the proposed HDC – KNN model for individually recognizing all the ten different classes. Besides, in the case of Digit 0, Digit 2, Digit 3, Digit 7, and Digit 8 classification it outperforms other proposed models for Bangla digit recognition in this work.

From Fig. 4.1 it can also be observed that the F1 Score of the HDC – KNN model for all the 10 handwritten English digits from digit 0 to digit 9 is 100%. These values indicate the perfection of the proposed HDC – KNN model for individually recognizing all the ten different English digit classes.

To assess the performance of the projected HDC – KNN structure in case of recognizing all the 10 handwritten Bangla and English digits, the True Positive Rate (TPR), and False Positive Rate (FPR) are premeditated and plotted in the graph for proper evaluation that is revealed in Fig. 4.2, and Fig. 4.3.

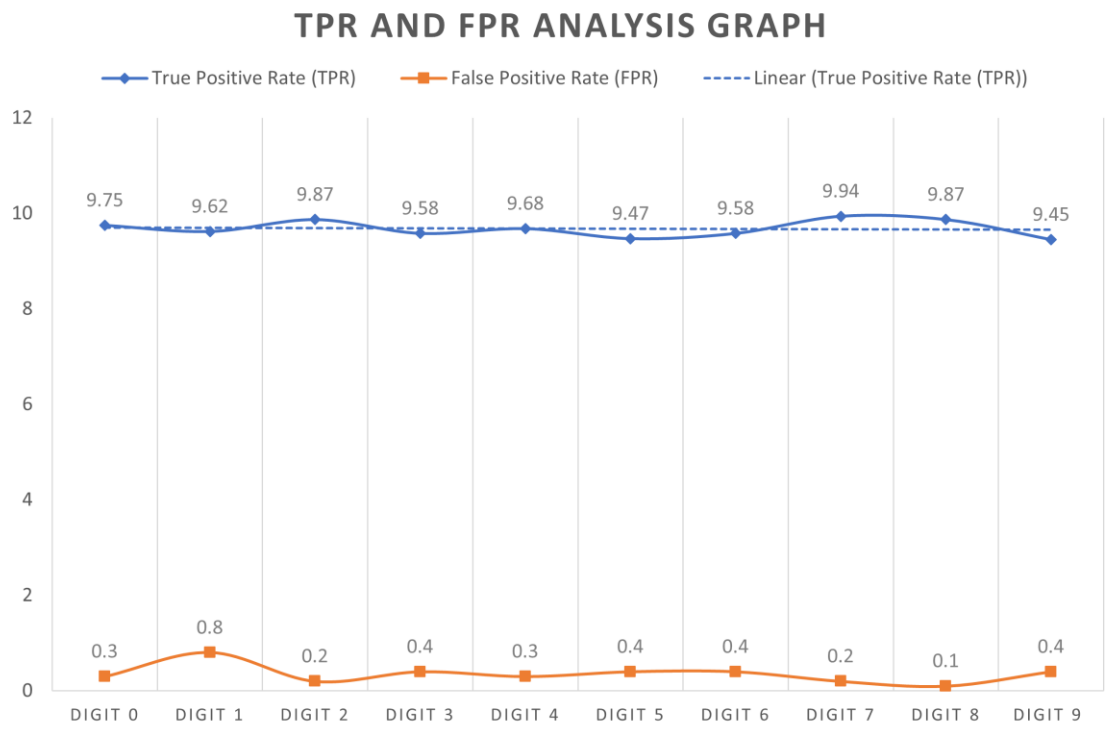


Fig. 4.2: TPR and FPR analysis graph for ten different Bangla digit categories

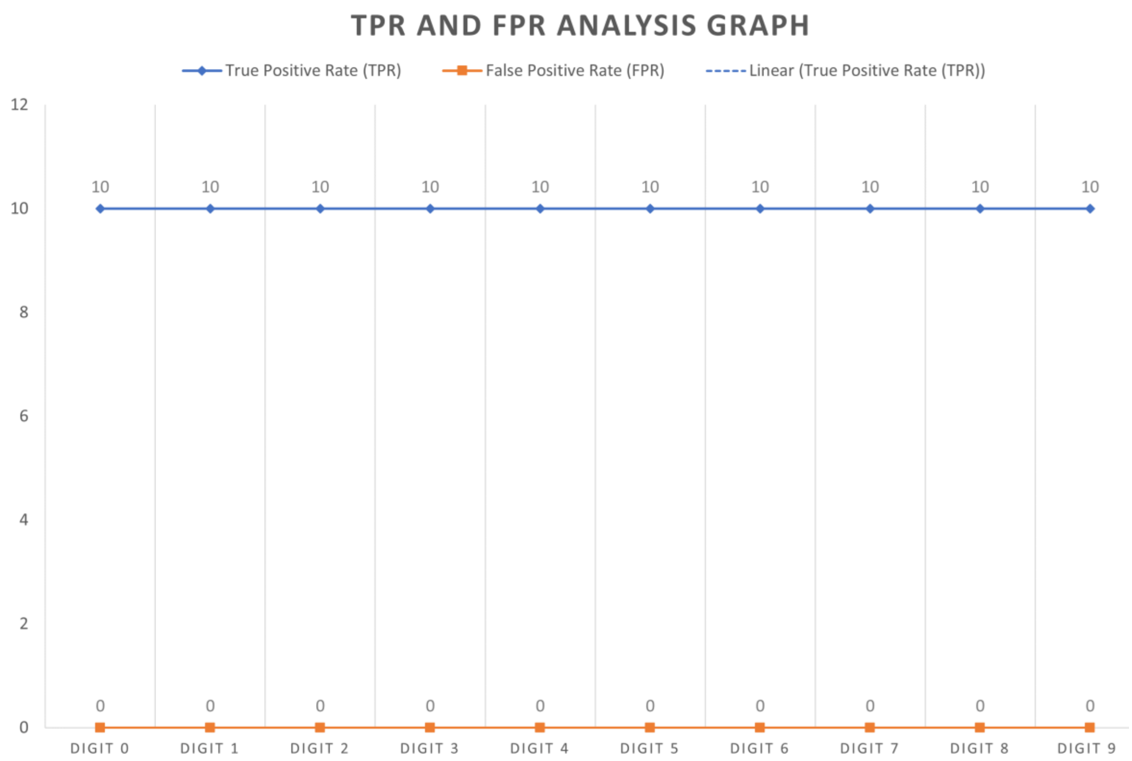


Fig. 4.3: TPR and FPR analysis graph for ten different English digit categories

Here, the explanation of TPR, or true positive rate, is given as the proportion of tasters in a dataset that the predictive model properly classifies as being in the positive category. The value of TPR can be premeditated by using the following formula [107]-[108].

$$TPR = \frac{TP}{TP + FN} \qquad (4.2)$$

In the formula, TP, or True Positive, stands for a taster that belongs to the positive class and is appropriately classified, whereas FN, or False Negative, stands for a taster that should belong to the positive class but is erroneously classified as a member of the negative class.

On the other hand, FPR, or the false positive rate, is demarcated as the proportion of samples from the negative class in the dataset that were incorrectly projected to be in the positive class. The Value of FPR can be premeditated by using the following formula [109]-[110].

$$FPR = \frac{FP}{FP + TN} \qquad (4.3)$$

$$= 1 - \frac{TN}{FP + TN} \qquad (4.4)$$

$$= 1 - Specificity \qquad (4.5)$$

In the formula, FP, or False Positive, stands for a sample that should be categorized as negative but was mistakenly classified as positive, while TN, or True Negative, stands for a sample that should be classified as negative but was incorrectly classified.

The values of True Positive Rate (TPR), and False Positive Rate (FPR) shown in Fig. 4.2 and Fig. 4.3 are calculated by using the above-explained formula. In the scenario of handwritten Bangla digit recognition, from the Fig. 4.2 it is seen that, the TPR value of the HDC – KNN structure for all classes of Digit 0 to Digit 9 are 9.75, 9.62, 9.87, 9.58, 9.68, 9.47, 9.58, 9.94, 9.87 and 9.45 respectively. On the contrary, the FPR values of the HDC – KNN model for all classes of Digit 0 to Digit 9 are 0.3, 0.8, 0.2, 0.4, 0.3, 0.4, 0.4, 0.2, 0.1 and 0.4 respectively. In the case of handwritten English digit appreciation, from Fig. 4.3 it is seen that, the TPR value of the HDC – KNN model for all classes of Digit 0 to Digit 9 is 10. On the contrary, the FPR value of the HDC – KNN model for all classes of Digit 0 to Digit 9 is 0. From the values, it is identifiable that, The True Positive Rate (TPR) value for all classes is significantly higher than the value of False Positive Rate (FPR). This indicates that the HDC – KNN model is highly efficient and perfect for recognizing handwritten Bangla and English digits.

## 4.2 COMPARISON WITH EXISTING MODELS BASED ON BANGLALEKHA-ISOLATED DATASET UTILIZED BY DIFFERENT METHODS

For further evaluation, the two most highly efficient models: HDC and HDC – KNN proposed in this work are compared with several existing handwritten recognition models based on BanglaLekha-Isloated [83] dataset. These are shown in Fig. 4.4.

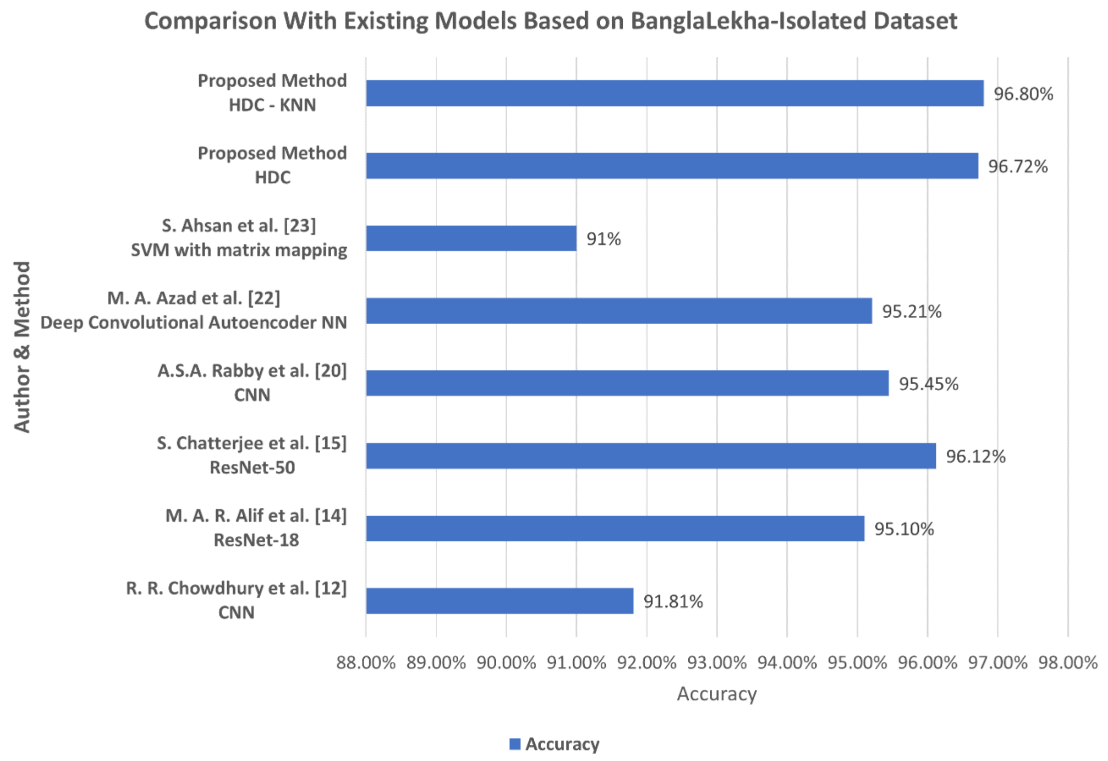


Fig. 4.4: Comparison with existing models based on BanglaLekha-Isloated dataset.

From Fig. 4.4, it can be observed that several existing papers [12], [14] – [15], [20], [22] – [23] work on BanglaLekha-Isolated dataset to categorize Bangla handwritten digits and shows decent outcomes. On the contrary, the two noteworthy models; the HDC and the HDC – KNN show an accuracy rate of 96.72% and 96.80% respectively, outclasses those existing models. This shows the high effectiveness of the proposed models of this research especially in the case of the HDC – KNN model.

## 4.3 COMPARISON WITH EXISTING MODELS BASED ON MNIST DATASET UTILIZED BY DIFFERENT METHODS

The developed HDC – KNN model in this work shows noteworthy results in the case of handwritten English digit appreciation. There are several other existing methods performed by various authors working on the MNIST [84] dataset in the scenario of English digit recognition which are compared with the proposed model in this work in Table 4.2.

Table 4.2: Comparison with existing models based on MNIST dataset utilized by different methods

| Author | Dataset | Method | Accuracy / Error Rate |
|---|---|---|---|
| X. Niu and C. Y. Suen et al. [30] | MNIST | CNN/SVM | 94.40 % |
| N. Saqib et al. [32] | MNIST | CNN using 'RMSprop' and 'ADAM' optimizer | 99.64 % |
| P. H. Jain et al. [33] | MNIST | CNN | 99.18 % |
| S. Ahlawat et al. [34] | MNIST | CNN | 99.89 % |
| A. Loquercio et al. [35] | MNIST | CNN | 00.17 % |
| K. Jarrett et al. [36] | MNIST | CNN | 00.53 % |
| L. Wan et al. [37] | MNIST | CNN | 00.21 % |

| | | | |
|---|---|---|---|
| T. G. Dietterich and G. Bakiri et al. [38] | MNIST | CNN | 00.19 % |
| A. A. Yahya et al. [39] | MNIST | CNN | 99.98 % |
| S. Ahlawat and A. Choudhary et al. [40] | MNIST | CNN/SVM | 99.28 % |
| S. Kulkarni and B. Rajendran et al. [41] | MNIST | SNN | 98.17 % |
| A. Garg et al. [42] | MNIST | CNN | 98.45 % |
| N. Schaetti et al. [43] | MNIST | CNN | 98.00 % |
| H. Zhao and H. Liu et al. [44] | MNIST | Multiple fusion CNN | 98.10 % |
| A. Tavanaei and A. S. Maida et al. [45] | MNIST | CNN | 98.00 % |
| H. Zhao and H. Liu et al. [46] | MNIST | CNN | 97.50 % |
| S. Ali et al. [47] | MNIST | CNN | 99.21 % |
| S. Ahlawat et al. [48] | MNIST | CNN | 99.89 % |
| S. Chakraborty et al. [49] | MNIST | CNN | 99.00 % |
| S. Ali et al. [50] | MNIST | CNN-ELM-DL4J | 99.80 % |
| W. Wang et al. [51] | MNIST | Homo-ELM | 97.00 % |
| D. Das et al. [52] | MNIST | ELM | 97.70 % |
| H. Malik and N. Roy et al. [53] | MNIST | ELM | 98.40 % |
| S. Ding et al. [54] | MNIST | CKELM | 96.80 % |
| X. Xu et al. [55] | MNIST | H-ELM<br>FCM-CNN-H-ELM | 99.10 %<br>98.70 % |
| **Proposed Method** | **MNIST** | **HDC-KNN** | **100%** |

From above Table 4.2, it can be observed that several existing papers worked on the MNIST dataset to categorize handwritten English digits and show decent outcomes. On the contrary, the proposed HDC – KNN model shows an accuracy rate of 100%, outclasses those existing models. This expresses the high efficacy of the projected model of this research especially in the scenario of handwritten English digit recognition.

## 4.4 COMPARISON WITH EXISTING MODELS BASED ON DIFFERENT DATASET UTILIZED BY DIFFERENT METHODS

As the developed Hybrid Dilated CNN (HDC) concept is a newer technique in the segment of image classification fewer models exist to date and in the arena of handwritten digit appreciation there is a lacking of notable works implementing this technique.

So, to prove the perfection of the unique HDC and HDC – KNN model developed in this work, a comparison with some different existing available HDC models evaluated on different data types is shown in Table 4.3.

Table 4.3: Evaluation of the projected model concerning similar architecture types based on different datasets

| Author | Dataset | Method | Computation Time | Accuracy |
|---|---|---|---|---|
| Z. Wang et al. [88] | SZ fMRI EPI | Dilated 3D CNN | - | 82.7% |
| Y. Lin and J. Wu [89] | HAR | Multichannel Dilated CNN | 15 minutes | 95.49% |
| L. Yin et al. [90] | CIFAR10 & CIFAR100 | Dilated Convolution + DenseNet and DPRN | - | 83.98% & 51.19% |
| R. Liu et al. [56] | SALINAS | HDC | 4.3969 s | 91.19% |
| X. Lei et al. [82] | Wide-band remote sensing image | HDC | 1248.30 s | 94.40% |
| **Proposed Method** | **BanglaLekha-Isolated** | **HDC** | **55 seconds** | **96.72%** |
| **Proposed Method** | **BanglaLekha-Isolated** | **HDC - KNN** | **55 seconds** | **96.80%** |
| **Proposed Method** | **MNIST** | **HDC - KNN** | **127 seconds** | **100%** |

From Table 4.3, it can be observed that the proposed unique HDC model in this work consumes 55 seconds of computation time and shows a significant accuracy of 96.72 %. However, another proposed model in this work, a fusion of Hybrid Dilated Convolution and K-Nearest Neighbors (KNN); the HDC – KNN model outclasses the all-other models shown in the table and shows an accuracy rate of 96.80 % for Bangla digit recognition and 100 % for English digit recognition consuming 55 seconds and 127 seconds respectively. Thus, the HDC – KNN model proves itself a standard and perfect architecture for detecting handwritten Bangla and English digits.

# Chapter 5: CONCLUSIONS

*This chapter explains the scope of future research on this topic as well as a summary and evaluation of the works completed in this research. In addition, the major implications and limitations of this research are also mentioned in this chapter.*

## 5.1 CONCLUSION

This research presents a unique Hybrid Dilated CNN (HDC) and HDC – KNN architecture to detect handwritten Bangla and English digits. Firstly, a custom CNN architecture is designed in this work. One of the major issues of the CNN architecture is that, if the layer increases for a more understandable learning system, it tends to consume more computation time due to regular feature map-generating techniques. Besides, CNN at the learning stage tends to ignore the spatial similarity of adjacent regions in dataset images. So, to solve the above-described problem, a Dilated CNN model is proposed in this work. It significantly reduces the computation time which is 11 seconds less than the custom CNN model in case of Bangla digits. However, due to the Dilated CNN model's feature extracting technique, it inserts holes in the traditional convolutional layer to upsurge the receptive field. So, the kernel is not continuous and creates a 'gridding' problem. Thus, the situation derives into lesser useful feature extraction and finally affects in overall recognition accurateness of the system. In this work, an improved HDC model is suggested to address the aforementioned issue. Besides, the writing style of handwritten digits shares some similar but important characteristics with hyperspectral images, which CNN or Dilated CNN can't easily obtain but HDC has been shown to do. The model overcomes the all previously mentioned problems and shows an accuracy of 96.72% consuming 55 seconds in the scenario of handwritten Bangla digits. The Precision, Recall, and F1 Score for individual classes also show great and effective outcomes. Then, this work also focuses on machine learning processes for enhancing the handwritten digit recognition system. With the proposed HDC model, two well-known machine learning systems, Support Vector Machine (SVM) and K-Nearest Neighbors (KNN), are implemented. The SVM and KNN are trained for classification tasks using the HDC model's extracted features. SVM's Structural Risk Minimization technique is reluctant to prevail over HDC’s improved Empirical Risk Minimization technique. However, KNN’s Euclidian distance measurement technique for classification shows a significant outcome and outperforms the HDC model. The HDC – KNN model expresses an accuracy rate of 96.80% for the Bangla digit and 100% for the English digit. The Precision, Recall, and F1 scores for individual classes also indicate the perfection of the model. True Positive Rate (TPR), and False Positive Rate (FPR) for all the 10-digit classes of the HDC – KNN model also indicate the high capability of the model to categorize the handwritten Bangla and English digits. These results and described parameters prove that the proposed HDC model can capture more features and surpass the difficulties of CNN and Dilated CNN architecture. Therefore, the implementation of HDC and KNN by developing the hybrid HDC – KNN model proposed in this research proves itself as a state-of-the-art handwritten Bangla and English digit recognition method.

## 5.2 LIMITATIONS OF THE STUDY

In the field of digit recognition or image classification CNN is the most widely used algorithm. Particularly, in the segment of digit appreciation there is a lot of scope for contribution. These are dataset arrangement, dataset image pre-processing, feature extraction technique, and classification. The majority of the previously done work mainly focused on a specific segment from these.

Moreover, this work tries to cover every aspect of these fields. For this reason, there are some limitations in different segments of this research. These are mentioned below:

i. The two datasets used in this work; cover a large type of sample of human writing. This work only considers these samples. The samples can be more diversified.

ii. The proposed Dilated CNN model in this work can reduce computation time. However, it significantly decreases the accuracy in case of small image size i.e. handwritten dataset samples.

iii. The proposed Hybrid Dilated CNN (HDC) model's dilation rate design technique indicates a fixed number of layers based on image size.

iv. Fewer data samples were utilized in the case of the handwritten Bangla digit dataset compared to the English digit dataset.

## 5.3 RECOMMENDATION FOR FURTHER STUDY

Developing a structure for handwritten Bangla and English digit appreciation was the primary goal of this endeavor. The following areas can be taken into consideration to resolve the limitations of this research and enhance system performance.

- The images of datasets can be converted into more diversified and complex forms for training procedures.
- There is still scope to work on the Hybrid Dilated CNN (HDC) model's dilation rate design by following the research done in this work.
- Developing a dilation rate design formula for a greater number of layers can be considered.

- Parameter change of the SVM and KNN algorithm can be more briefly and precisely examined.
- In the case of the KNN algorithm for classification; Manhattan Distance and Cosine Distance can be considered to utilize for distance calculation.
- Besides, in the segment of classification, several other Machine Learning processes can be utilized maintaining proper procedures to observe the change in results.

## 5.4 IMPLICATIONS

The following are a few possible implications of this research:

- Improved Pre-processing Method: The suggested pathway in this work in the section on pre-processing can practically be utilized for dataset arrangement tasks. The method shows a systematic framework for an ideal pre-processing environment. This can help the training system to run smoothly.
- Modernized Feature Extraction Technique: In the arena of handwritten digit classification, this work designed and developed an effective way for extracting features from datasets based on different writing styles, fonts, curves, arrangements, etc. This technique can be widely implementable for character or sentence classification based on handwritten datasets.
- HDC Model Application: The proposed dilation rate design of the Hybrid Dilated CNN (HDC) model is mathematically proven in this work to solve the 'gridding' and 'pixel loss' drawbacks of the Dilated CNN model. Thus, this model's design technique can be

implemented in complex machine classification problem solving i.e. hyperspectral image classification.

- Hybrid Machine Learning Classification: This research shows a way to integrate the Hybrid Dilated CNN (HDC) model with popular machine learning algorithms i.e. SVM, KNN, etc. Extracted features from the HDC model can be transferrable to machine learning algorithms for classification tasks. This technique can be utilized for handwritten word or sentence categorization.

- Advanced Embedded Systems: The research conducted here has resulted in the development of advanced handwritten digit appreciation algorithms that may be used in embedded systems for an eclectic range of applications, such as smart devices and Internet of Things gadgets. Because these systems often require fast and accurate object recognition to function effectively.

- Effective User Interface Design: More intuitive and user-friendly interfaces in electronic devices like tablets, smartphones, and digital notepads are made possible by handwritten digit recognition. Enabling natural input techniques enhances the entire user experience. For this user-friendly interface design purpose, the method developed by this research can be greatly applied.

- Advanced Security Systems: Secure electronic systems use handwritten digit recognition to verify signatures and authenticate handwritten passwords. This improves the security of electronic transactions and access control systems. In this instance, the technique that the study has established can be considerably implemented for more perfection.

# References


[1] N. Alrobah and S. Albahli, "A hybrid deep model for recognizing arabic handwritten characters," in *IEEE Access*, vol. 9, pp. 87058-87069, 2021, doi: 10.1109/ACCESS.2021.3087647.

[2] A. Baldominos, Y. Saez, and P. Isasi, "A Survey of Handwritten Character Recognition with MNIST and EMNIST," *Applied Sciences*, vol. 9, no. 15, p. 3169, Aug. 2019, doi: https://doi.org/10.3390/app9153169.

[3] A. S. Alqahtani, A. N. Madheswari, A. Mubarakali, and P. Parthasarathy, "Secure communication and implementation of handwritten digit recognition using deep neural network," *Optical and Quantum Electronics*, vol. 55, no. 1, Nov. 2022, doi: https://doi.org/10.1007/s11082-022-04290-7.

[4] R. Dixit, R. Kushwah, and S. Pashine, "Handwritten Digit Recognition using Machine and Deep Learning Algorithms," *International Journal of Computer Applications*, vol. 176, no. 42, pp. 27–33, Jul. 2020, doi: https://doi.org/10.5120/ijca2020920550.

[5] F. Y. Chin, K. H. Lem, and K. M. Wong, "Improving handwritten digit recognition using hybrid feature selection algorithm," *Applied Computing and Informatics*, Jul. 2022, doi: https://doi.org/10.1108/aci-02-2022-0054.

[6] N. Anusha, S. Jain, T. Shetty, and K. S. Bhat, "Digital Recognition of Handwritten Digits Using Convolutional Neural Networks," *Algorithms for intelligent systems*, pp. 351–366, Jan. 2022, doi: https://doi.org/10.1007/978-981-19-3311-0_30.

[7] N. Kadhre, K. S. Desale, R. More, and C. Palghadmal, "Recognition of Handwritten Digits Using Neural Networks: A Review," *Lecture notes in networks and systems*, pp. 783–792, Nov. 2022, doi: https://doi.org/10.1007/978-981-19-3148-2_67.

[8] Wikipedia Contributors, "Handwriting recognition," *Wikipedia*, May 02, 2019. https://en.wikipedia.org/wiki/Handwriting_recognition

[9] P. Baheti, "Handwriting Recognition: Definition, Techniques & Uses," *www.v7labs.com*, Dec. 02, 2022. https://www.v7labs.com/blog/handwriting-recognition-guide

[10] K. He, X. Zhang, S. Ren and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, 2016, pp. 770-778, doi: 10.1109/CVPR.2016.90.

[11] M. J. Hasan, M. F. Wahid, M. S. Alom and M. M. A. Mia, "A new state of art deep learning approach for bangla handwritten digit recognition using svm classifier," in *2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, Kharagpur, India, 2020, pp. 1-6, doi: 10.1109/ICCCNT49239.2020.9225367.

[12] R. R. Chowdhury, M. S. Hossain, R. ul Islam, K. Andersson and S. Hossain, "Bangla handwritten character recognition using convolutional neural network with data augmentation," in *2019 Joint 8th International Conference on Informatics, Electronics & Vision (ICIEV) and 2019 3rd International Conference on Imaging, Vision & Pattern Recognition (icIVPR)*, Spokane, WA, USA, 2019, pp. 318-323, doi: 10.1109/ICIEV.2019.8858545.

[13] M. R. Bashar, M. A.F.M. R. Hasan, M. A. Hossain and D. Das, "Handwritten bangla numerical digit recognition using histogram technique," *Asian Journal of Information Technology*, vol. 3, pp. 611-615, 2004.

[14] M. A. R. Alif, S. Ahmed, and M. A. Hasan, "Isolated bangla handwritten character recognition with convolutional neural network," in *20th International Conference of Computer and Information Technology (ICCIT), Dhaka, Bangladesh,* pp. 1-6, 2017. doi: 10.1109/ICCITECHN.2017.8281823.

[15] S. Chatterjee, R. K. Dutta, D. Ganguly, K. Chatterjee, and S. Roy, "Bengali handwritten character classification using transfer learning on deep convolutional neural network," *https://arxiv.org*, https://arxiv.org/abs/1902.11133 (accessed Jul. 25, 2023). https://doi.org/10.48550/arXiv.1902.11133.

[16] S. M. A. Hakim and Asaduzzaman, "Handwritten bangla numeral and basic character recognition using deep convolutional neural network," in *2019 International Conference on Electrical, Computer and Communication Engineering (ECCE),* Cox'sBazar, Bangladesh, 2019, pp. 1-6, doi: 10.1109/ECACE.2019.8679243.

[17] H. Zunair, N. Mohammed and S. Momen, "Unconventional wisdom: A new transfer learning approach applied to bengali numeral classification," in *2018 International Conference on Bangla Speech and Language Processing (ICBSLP)*, (Sylhet, Bangladesh), pp. 1-6, 2018. doi: 10.1109/ICBSLP.2018.8554435.

[18] O. Paul, "Image pre-processing on NumtaDB for bengali handwritten digit recognition," in *2018 International Conference on Bangla Speech and Language Processing (ICBSLP)*, (Bangladesh), pp. 1-6, Sylhet, 2018. doi: 10.1109/ICBSLP.2018.8554910.

[19] R. Basri, M. R. Haque, M. Akter, and M. S. Uddin, "Bangla handwritten digit recognition using deep convolutional neural network," in *Proceedings of the International Conference on Computing Advancements (ICCA 2020)*, (New York, NY, USA, Article 56), pp. 1–7, Association for Computing Machinery, 2020. https://doi.org/10.1145/3377049.3377077.

[20] A. S. A. Rabby, S. Haque, S. Islam, S. Abujar, and S. A. Hossain, "BornoNet: Bangla handwritten characters recognition using convolutional neural network," *Procedia Computer Science*, Vol. 143, 2018, Pages 528-535, ISSN 1877-0509, https://doi.org/10.1016/j.procs.2018.10.426.

[21] S. Jaman *et al.*, "Bangla handwritten: a comparative study among single, numeral, vowel modifier, and compound characters recognition using cnn," in *2022 13th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, Kharagpur, India, pp. 1-8, 2022, doi: 10.1109/ICCCNT54827.2022.9984562.

[22] M. A. Azad, H. S. Singha and M. M. H. Nahid, "Bangla handwritten character recognition using deep convolutional autoencoder neural network," in *2020 2nd International Conference on Advanced Information and Communication Technology (ICAICT)*, Dhaka, Bangladesh, pp. 295-300, 2020. doi: 10.1109/ICAICT51780.2020.9333472.

[23] S. Ahsan, S. T. Nawaz, T. B. Sarwar, M. S. U. Miah, and A. Bhowmik, "A machine learning approach for bengali handwritten vowel character recognition," *IAES International Journal of Artificial Intelligence (IJ-AI)*, vol. 11, no. 3, p. 1143, 2022. doi: 10.11591/ijai.v11i3.pp1143-1152.

[24] S. Bhowmik, S. Malakar, R. Sarkar *et al.*, "Off-line bangla handwritten word recognition: a holistic approach," *Neural Comput & Applic*, vol. 31, pp. 5783–5798, 2019. https://doi.org/10.1007/s00521-018-3389-1.

[25] R. Pramanik, and S. Bag, "Handwritten bangla city name word recognition using cnn-based transfer learning and fcn," *Neural Comput & Applic*, vol. 33, pp. 9329–9341, 2021. https://doi.org/10.1007/s00521-021-05693-5.

[26] A. Hazra, P. Choudhary, S. Inunganbi, et al., "Bangla-meitei mayek scripts handwritten character recognition using convolutional neural network," *Appl Intell*, vol. 51, pp. 2291–2311, 2021.

[27] S. Bag, G. Harit and P. Bhowmick, "Recognition of bangla compound characters using structural decomposition," *Pattern Recognition*, vol. 47, Issue 3, Pages 1187-1201, 2014, ISSN 0031-3203, https://doi.org/10.1016/j.patcog.2013.08.026.

[28] F. Siddique, S. Sakib and M. A. B. Siddique, "Recognition of handwritten digit using convolutional neural network in python with tensorflow and comparison of performance for various hidden layers," in *2019 5th International Conference on Advances in Electrical Engineering (ICAEE)*, Dhaka, Bangladesh, pp. 541-546, 2019, doi: 10.1109/ICAEE48663.2019.8975496.

[29] P. K. Singh, R. Sarkar and M. Nasipuri, "A study of moment based features on handwritten digit recognition", *Applied Computational Intelligence and Soft Computing*, vol. 2016, Article ID 2796863, 17 pages, 2016. https://doi.org/10.1155/2016/2796863

[30] X.-X. Niu, C. Y. Suen, "A novel hybrid CNN–SVM classifier for recognizing handwritten digits," *Pattern Recognition*, Volume 45, Issue 4, Pages 1318-1325, 2012, ISSN 0031-3203, https://doi.org/10.1016/j.patcog.2011.09.021.

[31] Y. S. Chernyshova, A. V. Sheshkus and V. V. Arlazarov, "Two-step cnn framework for text line recognition in camera-captured images," *IEEE Access*, vol. 8, pp. 32587-32600, 2020, doi: 10.1109/ACCESS.2020.2974051.

[32] N. Saqib, K. F. Haque, V. P. Yanambaka, and A. Abdelgawad, "Convolutional-neural-network-based handwritten character recognition: An approach with massive multisource data," *Algorithms*, vol. 15, no. 4, p. 129, Apr. 2022. doi:10.3390/a15040129

[33] P. H. Jain *et al.*, "Artificially intelligent readers: An adaptive framework for original handwritten numerical digits recognition with OCR methods," *Information*, vol. 14, no. 6, p. 305, May 2023. doi:10.3390/info14060305

[34] S. Ahlawat, A. Choudhary, A. Nayyar, S. Singh, and B. Yoon, "Improved handwritten digit recognition using convolutional neural networks (cnn)," *Sensors*, vol. 20, no. 12, p. 3344, Jun. 2020. doi:10.3390/s20123344

[35] A. Loquercio, F. D. Torre, and M. Buscema, "Computational eco-systems for handwritten digits recognition," *arXiv.org*, https://arxiv.org/abs/1703.01872 (accessed Mar. 5, 2024).

[36] K. Jarrett, K. Kavukcuoglu, M. Ranzato, and Y. LeCun, "What is the best multi-stage architecture for object recognition?," *NYU Scholars*, https://nyuscholars.nyu.edu/en/publications/what-is-the-best-multi-stage-architecture-for-object-recognition (accessed Mar. 5, 2024).

[37] L. Wan, M. Zeiler, S. Zhang, Y. L. Cun, and R. Fergus, "Regularization of neural networks using DropConnect," *PMLR*, https://proceedings.mlr.press/v28/wan13.html (accessed Mar. 5, 2024).

[38] T. G. Dietterich and G. Bakiri, “Solving multiclass learning problems via error-correcting output codes,” *Journal of Artificial Intelligence Research*, https://www.jair.org/index.php/jair/article/view/10127 (accessed Mar. 5, 2024).

[39] A. A. Yahya, J. Tan, and M. Hu, “A novel handwritten digit classification system based on convolutional neural network approach,” *Sensors*, vol. 21, no. 18, p. 6273, Sep. 2021. doi:10.3390/s21186273.

[40] S. Ahlawat and A. Choudhary, “Hybrid CNN-SVM classifier for handwritten digit recognition,” *Procedia Computer Science*, https://www.sciencedirect.com/science/article/pii/S1877050920307754 (accessed Mar. 5, 2024).

[41] S. Kulkarni and B. Rajendran, “Spiking neural networks for handwritten digit recognition-supervised learning and network optimization,” *Neural networks: the official journal of the International Neural Network Society*, https://pubmed.ncbi.nlm.nih.gov/29674234/ (accessed Mar. 5, 2024).

[42] A. Garg, D. Gupta, S. Saxena and P. P. Sahadev, "Validation of random dataset using an efficient cnn model trained on mnist handwritten dataset," in *2019 6th International Conference on Signal Processing and Integrated Networks (SPIN)*, Noida, India, pp. 602-606, 2019. doi: 10.1109/SPIN.2019.8711703.

[43] N. Schaetti, M. Salomon and R. Couturier, "Echo state networks-based reservoir computing for mnist handwritten digits recognition," in *2016 IEEE Intl Conference on Computational Science and Engineering (CSE) and IEEE Intl Conference on Embedded and Ubiquitous Computing (EUC) and 15th Intl Symposium on Distributed Computing and Applications for Business Engineering (DCABES)*, Paris, France, pp. 484-491, 2016. doi: 10.1109/CSE-EUC-DCABES.2016.229.

[44] H. Zhao and H. Liu, “Multiple classifiers fusion and CNN feature extraction for handwritten digits recognition,” *Granular Computing*, vol. 5, no. 3, pp. 411–418, Feb. 2019. doi:10.1007/s41066-019-00158-6

[45] A. Tavanaei and A. S. Maida, "Multi-layer unsupervised learning in a spiking convolutional neural network," in *2017 International Joint Conference on Neural Networks (IJCNN)*, Anchorage, AK, USA, pp. 2023-2030, 2017. doi: 10.1109/IJCNN.2017.7966099.

[46] H. Zhao and H. Liu, “Algebraic fusion of multiple classifiers for handwritten digits recognition,” in *2018 International Conference on Wavelet Analysis and Pattern Recognition (ICWAPR)*, Jul. 2018. doi:10.1109/icwapr.2018.8521321.

[47] S. Ali et al., “An efficient and improved scheme for handwritten digit recognition based on convolutional neural network,” *SN Applied Sciences*, vol. 1, no. 9, Aug. 2019. doi:10.1007/s42452-019-1161-5.

[48] S. Ahlawat, A. Choudhary, A. Nayyar, S. Singh, and B. Yoon, “Improved handwritten digit recognition using convolutional neural networks (CNN),” *Sensors*, vol. 20, no. 12, p. 3344, Jun. 2020. doi:10.3390/s20123344.

[49] S. Chakraborty, S. Paul, R. Sarkar, and M. Nasipuri, “Feature map reduction in CNN for handwritten digit recognition,” *Advances in Intelligent Systems and Computing*, pp. 143–148, Sep. 2018. doi:10.1007/978-981-13-1280-9_14.

[50] S. Ali et al., “An effective and improved CNN-elm classifier for handwritten digits recognition and classification,” *Symmetry*, vol. 12, no. 10, p. 1742, Oct. 2020. doi:10.3390/sym12101742.

[51] W. Wang, Y. Gan, C.-M. Vong, and C. Chen, “Homo-elm: Fully homomorphic extreme learning machine,” *International Journal of Machine Learning and Cybernetics*, vol. 11, no. 7, pp. 1531–1540, Jan. 2020. doi:10.1007/s13042-019-01054-w.

[52] D. Das, D. R. Nayak, R. Dash, and B. Majhi, “An empirical evaluation of extreme learning machine: Application to handwritten character recognition,” *Multimedia Tools and Applications*, vol. 78, no. 14, pp. 19495–19523, Feb. 2019. doi:10.1007/s11042-019-7330-0.

[53] H. Malik and N. Roy, “Extreme learning machine-based image classification model using handwritten digit database,” *Advances in Intelligent Systems and Computing*, pp. 607–618, Sep. 2018. doi:10.1007/978-981-13-1822-1_57.

[54] S. Ding, L. Guo, and Y. Hou, “Extreme learning machine with kernel model based on deep learning,” *Neural Computing and Applications*, vol. 28, no. 8, pp. 1975–1984, Jan. 2016. doi:10.1007/s00521-015-2170-y.

[55] X. Xu, S. Li, T. Liang, and T. Sun, “Sample selection-based hierarchical extreme learning machine,” *Neurocomputing*, vol. 377, pp. 95–102, Feb. 2020. doi:10.1016/j.neucom.2019.10.013.

[56] R. Liu, W. Cai, G. Li, X. Ning and Y. Jiang, "Hybrid dilated convolution guided feature filtering and enhancement strategy for hyperspectral image classification," in *IEEE*

*Geoscience and Remote Sensing Letters*, vol. 19, pp. 1-5, 2022, Art no. 5508105, doi: 10.1109/LGRS.2021.3100407.

[57] CASIO Education, "A Brief History of Numbers: How 0-9 Were Invented," *Casio Calculators*, Aug. 2018. https://education.casio.co.uk/blog/a-brief-history-of-numbers/

[58] "Decimal | Definition, Example, & Facts," *Encyclopedia Britannica*. https://www.britannica.com/science/decimal

[59] IBM, "What Are Neural Networks?," *www.ibm.com*, 2023. https://www.ibm.com/topics/neural-networks

[60] R. Yamashita, M. Nishio, R. K. G. Do, and K. Togashi, "Convolutional Neural networks: an Overview and Application in Radiology," *Insights into Imaging*, vol. 9, no. 4, pp. 611–629, Jun. 2018, doi: https://doi.org/10.1007/s13244-018-0639-9.

[61] Wikipedia Contributors, "Convolutional neural network," *Wikipedia*, Feb. 27, 2019. https://en.wikipedia.org/wiki/Convolutional_neural_network

[62] V. E. Balas, B. K. Mishra, and R. Kumar, *Handbook of Deep Learning in Biomedical Engineering*. Academic Press, 2020.

[63] M. Mittal, R. R. Shah, and S. Roy, *Cognitive Computing for Human-Robot Interaction*. Academic Press, 2021.

[64] S. Liu, J. Mcgree, Z. Ge, and Y. Xie, *Computational and statistical methods for analysing big data with applications*. Amsterdam: Elsevier/Academic Press, 2016.

[65] Wikipedia Contributors, "Rectifier (neural networks)," *Wikipedia*, Aug. 05, 2024. https://en.wikipedia.org/wiki/Rectifier_(neural_networks) (accessed Sep. 19, 2024).

[66] "Rectified Linear Unit (ReLU) layer - MATLAB - MathWorks United Kingdom," *uk.mathworks.com*. https://uk.mathworks.com/help/deeplearning/ref/nnet.cnn.layer.relulayer.html

[67] F. Yu, and V. Koltun, "Multi-scale context aggregation by dilated convolutions,", 2016. https://doi.org/10.48550/arXiv.1511.07122

[68] "What is Dilated Convolution - Javatpoint," *www.javatpoint.com*, 2024. https://www.javatpoint.com/what-is-dilated-convolution (accessed Sep. 19, 2024).

[69] Y. Shan, H. Lu, and W. Lou, "A hybrid attention and dilated convolution framework for entity and relation extraction and mining," *Scientific Reports*, vol. 13, no. 1, Oct. 2023, doi: https://doi.org/10.1038/s41598-023-40474-1.

[70] Y. Wang, Y. Zhao, L. Wu, Y. Zhang, and J. Hu, "Hybrid dilated convolutional neural network for solving electromagnetic inverse scattering problems," *International Journal of RF and Microwave Computer-Aided Engineering*, vol. 32, no. 3, Dec. 2021, doi: https://doi.org/10.1002/mmce.23023.

[71] Z. Wang, Y. Zou, and P. X. Liu, "Hybrid dilation and attention residual U-Net for medical image segmentation," *Computers in Biology and Medicine*, vol. 134, p. 104449, Jul. 2021, doi: https://doi.org/10.1016/j.compbiomed.2021.104449.

[72] Y. Fang, Y. Li, X. Tu, T. Tan, and X. Wang, "Face completion with Hybrid Dilated Convolution," *Signal Processing Image Communication*, vol. 80, pp. 115664–115664, Feb. 2020, doi: https://doi.org/10.1016/j.image.2019.115664.

[73] Wikipedia Contributors, "Softmax function," *Wikipedia*, Nov. 29, 2019. https://en.wikipedia.org/wiki/Softmax_function

[74] J. Ren and H. Wang, *Mathematical Methods in Data Science*. Elsevier, 2023.

[75] "Support Vector Machine Algorithm," https://www.javatpoint.com, https://www.javatpoint.com/machine-learning-support-vector-machine-algorithm (accessed Jul. 6, 2023).

[76] D. W. Sun and D.-W. Sun, *Computer vision technology for food quality evaluation*. Amsterdam; Boston: Elsevier/Academic Press, 2008.

[77] S. Bhattacharyya, V. Snášel, D. Gupta, and A. Khanna, *Hybrid computational intelligence : challenges and applications*. Amsterdam: Academic Press, 2020.

[78] "K-Nearest Neighbor (KNN) Algorithm," https://www.javatpoint.com, https://www.javatpoint.com/k-nearest-neighbor-algorithm-for-machine-learning (accessed Jul. 6, 2023).

[79] A. Urso, A. Fiannaca, M. L. Rosa, V. Ravì, and R. Rizzo, "Data Mining: Prediction Methods," *Encyclopedia of Bioinformatics and Computational Biology*, pp. 413–430, 2019, doi: https://doi.org/10.1016/b978-0-12-809633-8.20462-7.

[80] R. C. Neath and M. S. Johnson, "Discrimination and Classification," *International Encyclopedia of Education*, pp. 135–141, 2010, doi: https://doi.org/10.1016/b978-0-08-044894-7.01312-9.

[81] P. Nadkarni, “Chapter 10 - Core Technologies: Data Mining and ‘Big Data,’” in *Clinical Research Computing*, P. Nadkarni, Ed., Academic Press, pp. 187–204. Available: https://doi.org/10.1016/B978-0-12-803130-8.00010-5

[82] X. Lei, H. Pan and X. Huang, "A dilated cnn model for image classification," in *IEEE Access*, vol. 7, pp. 124087-124095, 2019. doi: 10.1109/ACCESS.2019.2927169.

[83] M. Biswas, R. Islam, G. K. Shom, M. Shopon, N. Mohammed, S. Momen, and A. Abedin, “Banglalekha-isolated: a multi-purpose comprehensive dataset of handwritten bangla isolated characters,” *Data in Brief*, Volume 12, Pages 103-107, 2017, ISSN 2352-3409. https://doi.org/10.1016/j.dib.2017.03.035.

[84] Y. LeCun, C. Cortes, and C. Burges, “The mnist database,” *MNIST handwritten digit database*, http://yann.lecun.com/exdb/mnist/ (accessed Mar. 5, 2024).

[85] Wikipedia Contributors, “Median filter,” *Wikipedia*, Aug. 23, 2019. https://en.wikipedia.org/wiki/Median_filter

[86] A. F. Frangi, W. J. Niessen, K. L. Vincken, and M. A. Viergever, “Multiscale vessel enhancement filtering,” *Medical Image Computing and Computer-Assisted Intervention — MICCAI’98*, pp. 130–137, 1998, doi: https://doi.org/10.1007/bfb0056195.

[87] “Resize image - MATLAB imresize,” *www.mathworks.com*. https://www.mathworks.com/help/matlab/ref/imresize.html

[88] Z. Wang, Y. Sun, Q. Shen and L. Cao, "Dilated 3d convolutional neural networks for brain mri data classification," *IEEE Access*, vol. 7, pp. 134388-134398, 2019. doi: 10.1109/ACCESS.2019.2941912.

[89] Y. Lin and J. Wu, “A novel multichannel dilated convolution neural network for human activity recognition,” *Mathematical Problems in Engineering*, vol. 2020, pp. 1–10, Jul. 2020. doi:10.1155/2020/5426532

[90] L. Yin, P. Hong, G. Zheng, H. Chen, and W. Deng, “A novel image recognition method based on densenet and dprn,” *Applied Sciences*, vol. 12, no. 9, p. 4232, Apr. 2022. doi: 10.3390/app12094232.

[91] P. Wang, P. Chen, Y. Yuan, D. Liu, Z. Huang, X. Hou, and G. Cottrell, “Understanding convolution for semantic segmentation,” 2017. https://doi.org/10.48550/arXiv.1702.08502

[92] L. -C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected

crfs," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 4, pp. 834-848, 1 April 2018. doi: 10.1109/TPAMI.2017.2699184.

[93] S. Mallick and S. Nayak, "Number of parameters and tensor sizes in a convolutional neural network (cnn)," *LearnOpenCV*, https://learnopencv.com/number-of-parameters-and-tensor-sizes-in-convolutional-neural-network/ (accessed Feb. 20, 2024).

[94] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, pp. 273–297, 1995. https://doi.org/10.1007/BF00994018.

[95] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE Transactions on Information Theory*, vol. 13, no. 1, pp. 21-27, January 1967. doi: 10.1109/TIT.1967.1053964.

[96] Wikipedia Contributors, "Confusion matrix," *Wikipedia*, Oct. 22, 2019. https://en.wikipedia.org/wiki/Confusion_matrix

[97] "Compute confusion matrix for classification problem - MATLAB confusionmat," *www.mathworks.com*. https://www.mathworks.com/help/stats/confusionmat.html

[98] "Manhattan distance," *xlinux.nist.gov*. https://xlinux.nist.gov/dads/HTML/manhattanDistance.html

[99] "Taxicab geometry," *Wikipedia*, Jan. 21, 2022. https://en.wikipedia.org/wiki/Taxicab_geometry

[100] F. Szabo, *The Linear Algebra Survival Guide*. Academic Press, pp. 101–118, Jan. 2015.

[101] Wikipedia Contributors, "Euclidean distance," *Wikipedia*, Apr. 01, 2019. https://en.wikipedia.org/wiki/Euclidean_distance

[102] Wikipedia Contributors, "Precision and recall," *Wikipedia*, Apr. 19, 2019. https://en.wikipedia.org/wiki/Precision_and_recall

[103] D. L. Olson and D. Delen, *Advanced Data Mining Techniques*. Springer Science & Business Media, 2008.

[104] "Precision and Recall in Machine Learning - Javatpoint," *www.javatpoint.com*. https://www.javatpoint.com/precision-and-recall-in-machine-learning

[105] K. Akre, "F-score | Definition, Formula, & Facts | Britannica," *www.britannica.com*. https://www.britannica.com/science/F-score

[106] "F-score," *Wikipedia*, Mar. 12, 2021. https://en.wikipedia.org/wiki/F-score

[107] H. Wang and H. Zheng, “True Positive Rate,” *Encyclopedia of Systems Biology*, pp. 2302–2303, 2013, doi: https://doi.org/10.1007/978-1-4419-9863-7_255.

[108] Wikipedia Contributors, “Sensitivity and specificity,” *Wikipedia*, Jan. 19, 2019. https://en.wikipedia.org/wiki/Sensitivity_and_specificity

[109] H. Wang and H. Zheng, “False Positive Rate,” *Springer eBooks*, pp. 732–732, Jan. 2013, doi: https://doi.org/10.1007/978-1-4419-9863-7_224.

[110] “False positive rate,” *Wikipedia*, May 21, 2021. https://en.wikipedia.org/wiki/False_positive_rate