NCE Journal of Science and Engineering (NJSE), Vol. 2, June 2025

Data Retirement Tool (DRT) -A Novel Tool for Data Archiving

Manish Shrestha[1], Subash Panday[1,*], and Monica Regmy[1]
[1]IIMS College, Lincoln University Nepal, Hattisar, Kathmandu
[*]Correspondence: subash@nce.edu.np
Manuscript received December 23, 2024; accepted March 11, 2025

***Abstract*—Effective archiving of data and maintaining efficient performance is a crucial task for managing the exponential growth of data in the database domain. A data retirement tool (DRT), was developed to address this issue using a partitioning technique within the database. Initially implemented in a procedural language/ structured query language (PLSQL) environment, the tool was based on PLSQL and its specific packages. To extend the scope of study, the DRT tool was implemented in my structured query language (MySQL), and its efficiency was evaluated in this article. The performance of the algorithm was assessed under two conditions: one involving table partitioning during deletion and the other without it. Data removal was carried out at rates of 10%, 50%, and 90%, with the results illustrated in graphical form. The results were positive, consistent with the findings in the initial referenced article**

***Keywords*—Data archiving, Data Retirement tool (DRT), Near Replica Rapid Deletion (NRRD) algorithm, Partition Exchange, Procedural language/Structured Query Language (PLSQL), My Structures Query Language (MySQL).**

I. INTRODUCTION

Database plays a very vital role in every IT domain for proper storage and retrieval of data in a systematic manner. In the present context, the data are generated exponentially in every domain. As a result, the processing and retrieval of data from the database gets hampered. The increase in volume of data to be stored results in slower performance of the database operation [1]. Accessing large volumes of data concludes undesirably higher cost operation. Therefore, proper management of a huge volume of data is a must [2]. There are several approaches to deleting data for archiving, such as indexing [3], database sharding [4], and compression [5], among others. Furthermore, it is crucial to avoid any downtime in a database system that requires high availability. Hence, among the various available techniques for data deletion, partitioning has been employed in this article. This choice is driven by the experimental nature of the study, which builds upon an established work that utilized partitioning for data deletion in the archiving process. Moreover, the focus is placed on evaluating the efficiency of the Data Removal Tool (DRT) in a MySQL [6] environment. Previously, the referenced article introduced an innovative Data Retirement Tool (DRT) that enabled data removal without causing system downtime or increased workload. In comparison to the standard SQL Delete method, DRT demonstrated. superior performance in both deletion speed and reduced system load in Oracle. [7]Hence, considering the drawbacks of standard SQL delete operation and its widespread utilization, there is a strong demand for a new data retirement tool that can alleviate system load, reduce lock contention, and maintain database objects in an optimal state after data removal [8]. A practical solution involves integrating existing high availability and reorganization capabilities with partitioning features to develop such a tool [2]. In most database systems, partitioning techniques are frequently utilized to enhance SQL query performance and optimize resource utilization [9]. This same approach can also significantly improve the efficiency of delete operations, as partitioning helps isolate and manage large datasets, reducing the need for extensive scans during data deletion. By targeting specific partitions instead of the entire database, partitioning leads to faster and more effective data removal, making it a crucial strategy in optimizing both query and delete performance in relational databases, as Gartner et al. [10] Suggested, “bulk deletes can be implemented very efficiently by simply discarding a whole partition (including all indices of that partition).”

Similar to referenced article, the same algorithm has been used in this study as well. Step by step the algorithm has been implemented in this study. The process of the NRRD algorithm initiates with the creation of replica of a base table (The table which needs archiving). The whole process can be summarized into three main steps. First, it applies partition by in the table such that the whole table can be arranged into the partition of efficient/optimal querying process and for archiving process. There will be two partition created namely “Archive” partition and “Keep” partition. In this way there will be a vivid segregation between data to be retained and data to be archived. Since partition approach is available in every relational database management system (RDBMS) [11] it will allow efficient bulk deletion of the data during the archiving process which has very minimal impact on the system load. Second, there is the process of table reorganization through partition exchange. Finally, in third step, the algorithm captures the ongoing changes done in the main table such that it can be reflected in the replica table. In this way there will be no locking

done on main table. On the other hand, the replica table will act like the base table to perform data archiving.

Data Archiving and Deletion: In this paper, the process and concept of data deletion and archiving is same as the referenced paper. First of all there will be a clear segregation between the data that needs to be archived and the data that needs to be deleted. This is done via a partition flag. Thus achieved data will be stored in other secondary storage devices [2]. Secondly, deleting huge volume of data from a database table can take significant amount of time based on the volume of data. As a result the targeted table will not be available for other operation. Hence, instead the algorithm will be operated on the replica table which will ultimetaly help in avoiding system downtime issue.

In a database, the complexity of data deletion operation is based on the concept of transactions that most of the databases use to ensure the database integrity [10]. When delete operation is performed on a block of data then the database locks that particular data block until it is completely removed from the locked resources and the assoicated indices. When there is large number of indices in a table then data delete operation must delete data from required table and from all the associated indices. The deletion of data and deletion of data from indices must be done at the same time to maintain the system consistency. Hence, as a result of this the whole process of traditiona; delete operation can time huge amount of system time [12].

II. RELATED WORKS

First and foremost, this study is based on the concept and methodology of 'DRT- a Novel Tool for Data Archiving'. In the referenced paper the 'NRRD' algorithm was implemented in PLSQL environment along with its specific packages [7]. The effectiveness of this algorithm was yet to be checked on other RDBMS since the concept upon which the algorithm is based on is available in other RDBMS also. The only difference is that other RDBMS might not have the specific packages used in the referenced paper. For bulk deletion, Gartner, et al. [8] introduced a new algorithm for efficient bulk delete operation. However, the proposed algorithm requires extensive locks on archived data, and tables and switches all indices on these tables in offline mode. Ultimately, this resulted in database downtime. Similarly, Mohan [13] conducted a related study, demonstrating that data deletion can be simplified by minimizing the number of lock calls through record-level key-ranging locking. However, this method has a limitation as it is only effective for cursor-based deletion operations.

Lilja et al. [14] enhanced Mohan's approach by introducing a multigranular key-range locking protocol. This method allows bulk deletion to lock only a limited number of logical table fragments, reducing the cost of B-tree rebalancing. However, both approaches necessitate [13] [14] database downtime.

III. METHODOLOGY

This section involves the detail explanation of NRRD algorithm (Near Replica Rapid Deletion) used by DRT tool. First of all, a table is selected in which data archive needs to be done. Let us consider this table as an original/main table.

Step 1: Create a replica of original table: The replica/exact copy of the selected original table is made. Let the table be named as Replica table.

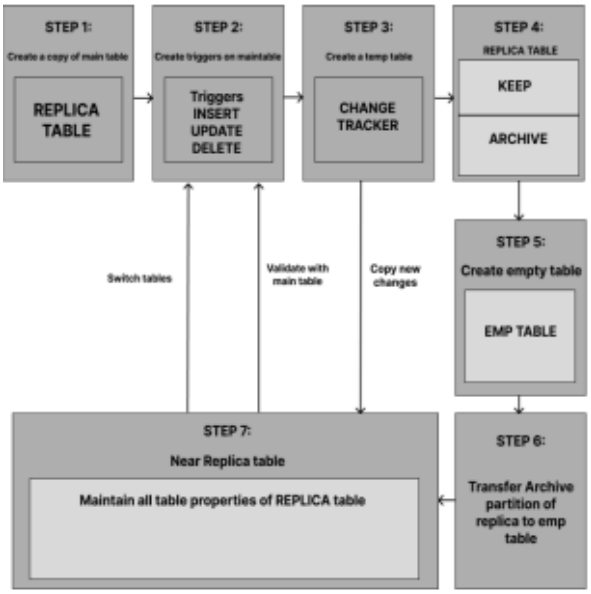

Fig. 1. Near Replica Rapid Deletion (NRRD) algorithm.

Step 2: Creation of triggers: Trigger are created in the main table such that every change that are made in the main table can be logged into a log table.

Step 3: Creation of a log table: A log table is specially designed for maintaining or storing the changes that are made in the main table during the processing of this algorithm in order to maintain the concurrency. Let the table be named as change tracker.

Step 4: Partitioning replica table: The replica table is modified to have to partition. 'Archive' and 'Keep' partition. 'Archive' partition is meant to store all the records of table in which the flag is archive. Similarly, the 'Keep' partition is meant to store all the records having 'Keep' partition.

Step 5: Creation of an empty table: Create an empty table named 'Emp_table'. It is created to store only archive date.

Step 6: Partition exchange or transfer: By using the partition exchange technique, transfer the archive partition to the Emp_table. (Emp_table has no partition)

Step 7: Synchronization with original table: During the execution of the algorithm, there might be some change done in the main table. Hence, in order to maintain the integrity of data and to reflect all the necessary change to the replica table, we need to check and make changes in replica table accordingly to make replica table an exact copy of main table.

Step 8: Future data management: When the necessary change fail to be reflected in the replica table then the whole process needs to be repeated. On the other if all the changes are reflected then the replica table is renamed to main table in which all the records would have keep flag only. I.e. Archive partition is empty. In future when the table start to get populated again with new data then such data are aligned to keep and archive partition on the basis of their flag.

In MySQL similar to referenced paper, CTAS approach is used to create the replica table. In order to maintain the concurrency, the replica table will be the base table for the algorithm to work on. Another important aspect of this algorithm is to maintain data integrity.

I.e. reflecting all the changes done in main table into the replica table. This is done after successful partition exchange operation. In partition exchange operation, all the data which needs to be archived will be shifted to another table and there will be only required data in the replica table. And this data must be exactly same as in the main table. And this process of synchronization will be done in step 7. Furthermore the process of synchronization is based on 'Change tracker' table. This table is responsible for storing logs of all the change done in the main table. Record will be inserted in the "Change tracker" table when the trigger of main table is triggered. In main table trigger such as "Insert", "Delete", "Update" are created such that insert, delete, update operation log can be logged into change tracker table. Finally, after the successful archiving and synchronization operation, the replica table will be renamed to be used as main table. Hence in this way NRRD algorithm will help in reducing system downtime when simulating traditional delete operation which required huge amount of time when deleting huge volume of data.

IV.RESULT AND DISCUSSION

As per the output obtained from the referenced paper, the results were positive in nature. The deletion of the huge volume of data took a significant amount of time whereas the NRRD algorithm was able to simulate the delete operation using the partition technique and perform the operation in very little time compared to the traditional way of bulk delete. Furthermore, the output also depends on the number of indexes the table has. The higher the number of indexes, the higher the execution time of the deletion operation and vice versa. However, after implementing the algorithm in the MySQL environment, there were some test cases where the results were negative and for rest of the test cases, the results were positive. Furthermore, in the test case environment where the test cases were conducted, ~1.5 million rows were considered. In the cases where there was 1 index & less percent of data to be deleted, the NRRD algorithm could not outperform the deletion operation. On the other hand, in the test cases where there were more number of index and more percent of data deletion was required the NRRD algorithm outperformed the deletion operation. Hence, analysing these outcomes and the provided scenario we can conclude that the NRRD algorithm performs as expected in the MySQL environment also. Furthermore, the referenced paper has discussed the synchronization of the replica table with respect to the original table. All the change done in original table must be reflected on the replica table. However, the paper has not discussed the amount of changes to be done and their synchronization with the original table. If a huge volume of data needs to be changed or updated then it may take a significant amount of time to do so. This may hamper the matrices that was gather. In contrast, if the amount of change that needs to be done is not huge in volume then it may not hamper the gathered matrices.

The experimental results were presented using graphs created with Microsoft Excel tool [15]. This tool was instrumental in illustrating the findings clearly and effectively, making them easier to interpret and understand. The graphs visually represent the outcomes, enhancing their clarity and appeal. Below are the graphs generated from the experiment performed.

For deleting 10% of data, the traditional delete method outperforms the NRRD algorithm in two cases

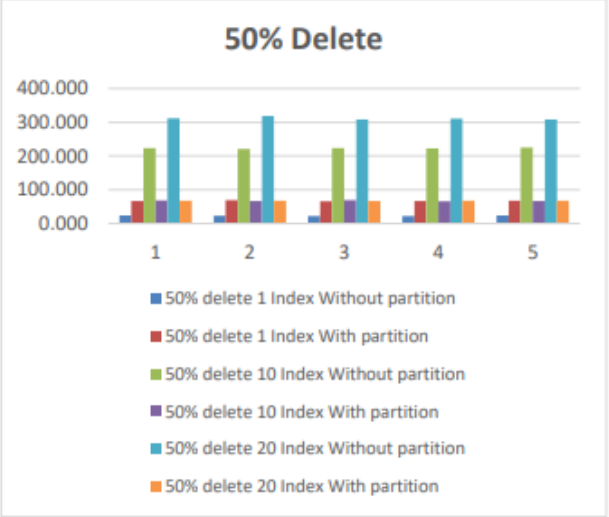

Fig. 2. 10% delete graph.

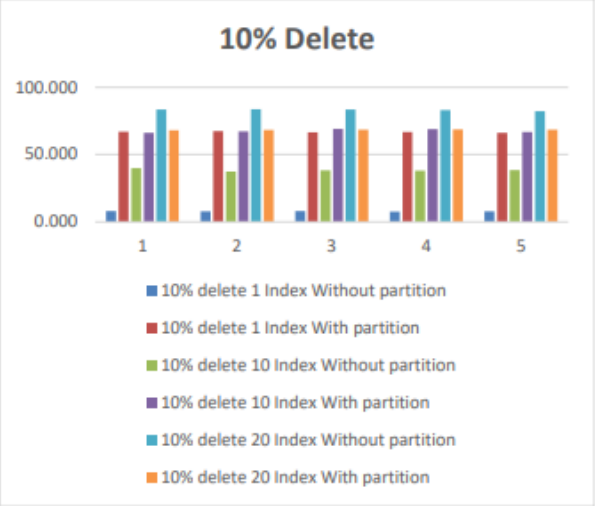

Fig. 3. 50% Delete Graph

1. 10% delete having 1 index.
2. 10% delete having 10 indexes.

In contrast for deleting 10% of data, the NRRD outperforms the traditional delete method in:

1. 10% delete having 20 indexes.

For deleting 50% of data, the traditional delete method outperforms the NRRD algorithm in:

1. 50% delete having 1 index.

In contrast for deleting 10% of data, the NRRD outperforms the traditional delete method in:

2. 50% delete having 10 indexes.
3. 50% delete having 20 indexes.

In contrast for deleting 10% of data, the NRRD outperforms the traditional delete method in:

2. 50% delete having 10 indexes.
3. 50% delete having 20 indexes.

For deleting 90% of data, the traditional delete method outperforms the NRRD algorithm in:

1. 50% delete having 1 index

In contrast for deleting 90% of data, the NRRD outperforms the

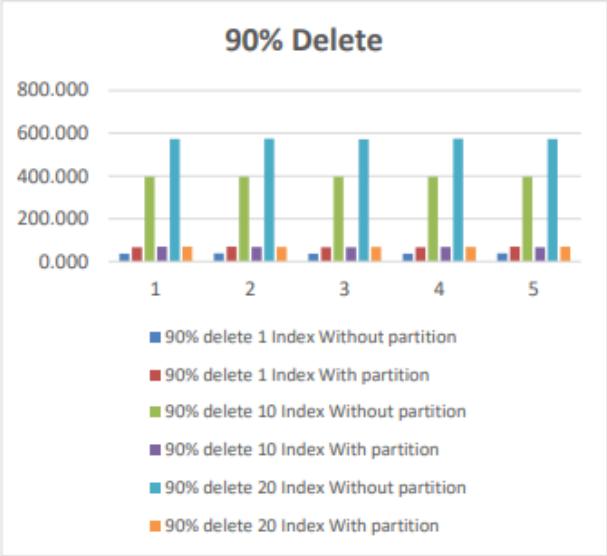

Fig. 4. 90% delete graph.

traditional delete method in:

4. 90% delete having 10 indexes.
5. 90% delete having 20 indexes.

Hence, based on the result thus obtained after the algorithm implementation in MySQL, it can be concluded that, the algorithm outperforms the traditional delete method when the volume of data is huge and when there are many indexes in the table. Therefore, even though negative results can be seen for few test cases in this study, it can be concluded that the NRRD algorithm also performs efficiently in MySQL environment.

ACKNOWLEDGMENT

I am deeply thankful to my advisor, Asst. Prof. Subash Panday, for their exceptional guidance, motivation, and unwavering support throughout this study. Their constructive advice and expertise were critical in shaping the direction of my research. I also sincerely thank Kelvin Williams, Yan LI, and Lorne Olfman for their groundbreaking work on the algorithm that forms the core of this study. Their paper, "DRT – A Novel Tool for Data Archiving," served as a vital reference and provided significant insights that greatly contributed to the progress and completion of this research.

REFERENCES

[1] M. L. Gillenson, Fundamentals of database management systems, John wiley & Sons, Inc., 2023.

[2] K. C. a. D. L. Sheila Childs, "Magic Quadrant for Enterprise Information Archiving," *Gartner, Inc.,* 29 October 2010.

[3] M. G. Cecilia CIOLOCA, "Increasing Database Performance using Indexes," *Database Systems Journal,* vol. II, pp. 13- 22, 2011.

[4] B. M. Abdelhafiz and M. Elhadef, "Sharding Database for Fault Tolerance and Scalability of Data," *IEEE Xplore,* 25 February 2021.

[5] D. A. L. a. D. S. HIRSCHBERG, "Data Compression," *ACM Computing SUWSYS,* Vols. Vol, 19, pp. 261-296, Spetember 1987.

[6] S. Suehring, MySQL Bible, 2001.

[7] Y. L. L. O. Kevin Williams, "DRT: A Novel Tool For Data Archiving," *IEEE Software,* vol. 38, no. 2, pp. 88-95, 27 02 2020.

[8] A. K. D. K. B. Z. A. G¨artner, "Efficient Bulk Deletes in Relational Databases," *Proceedings 17th International Conference on Data Engineering,* 07 April 2001.

[9] Y. Alsultanny, "Database management and partitioning to improve database processing performance," *Journal of Database Marketing & Customer Strategy Management,* vol. Volume 17, pp. 271-276, 2010.

[10] V. S. T. V. R. V. B. J. S. L To, Oracle Database High Availability Best Practices 11g release 2 (11.2) E10803-02.

[11] K. K. P. P. N. C. Mayur Sawant, "Database Partitioning: A Review Paper," *International Journal of Innovative Technology and Exploring Engineering (IJITEE),* vol. Vol. 3, no. Issue. 5, October 2013.

[12] S. Mukherjee, "Indexes in Microsoft SQL Server," *Arxiv.*

[13] C.Mohan, "An efficient method for performing record deletion and updates using index scans," *In proceedings of the 28th international conference on very Large Data Bases,* pp. pp. 940- 949, 2002.

[14] R. S. S. S. E. S.-S. Timo Lilja, "Online Bulk Deletion," *IEEE 23RD Internaltional Conference on Data Engineering,* pp. pp. 956-965, 2007.

[15] D. Kraus, "Consolidated data analysis and presentation using an open-source add-in for the Microsoft Excel® spreadsheet software.," *Medical Writing,* pp. 25-28, 2014.

NCE Journal of Science and Engineering (NJSE), Vol. 2, June 2025

Offline Rañjanā Lipi Handwritten Character Recognition Using CNN Model

Dinesh Maharjan[1] and Balkrishna Bal[2]
[1]Pokhara University, Kathmandu, Nepal
[2]Nepal College of Information Technology, Lalitpur, Nepal
[*]Correspondence: dinesh.maharjan@gmail.com and niranjan@ncit.edu.np
Manuscript received January 8, 2025; accepted April 20, 2025

***Abstract*—This research is focus on "Offline Rañjanā Lipi Handwritten Character Recognition (ORLHCR)" using a (CNN) Convolutional Neural Network and the development of Rañjanā Lipi Dataset (RLD). About 40 students voluntarily contribute to develop 6792 isolated character images of 62 classes. The developed dataset does not contain any data augmented data. Based on this RLD, the data's are divided into 75% Training, 25% Test set, and from Test set it is further split into 50-50 % i.e. 50% test and 50% validation set. Vgg19, InceptionV3, and ResNet50 CNN models are implemented to acquire the accuracy of ORLHCR. The test results delivers an accuracy rate of 96.82% by RestNet50 from RLD by tuning learning parameters only.**

***Keywords*—Convolutional Neural Network (CNN), handwritten character recognition, deep learning, Optical Character Recognition (OCR), hyper parameter - learning**

I. INTRODUCTION

HANDWRITTEN character recognition is one of the hot and interesting topic nowadays. Many researcher have worked on character recognition in Devanagari script but not in the field of Rañjanā Lipi/script. No digitization of this script found on the digital platform. The making of dataset of Rañjanā Lipi is a quite challenging job. The Rañjanā Lipi contain 16 vowels, 36 consonant and 0-9 digits. Compared to other scripts, the writing style of Rañjanā Lipi has a thick lining and beauty contains of this. The thick line of these scripts is to mark the paper for a longer duration. Rañjanā Lipi was used by inhabitant's people of Nepa Valley from Nepal to Tibet. The Lipi was developed during the 11th century which is Abugida's writing style. Many scripts were developed during middle ages from Nepal Bhasa such as Brahmi script, Prachalit script, Rañjanā Lipi/script, Bhujinmola script, Kunmol script, Kwenmol script, Golmol script, Pachumol script, Hinmol script, and Litumol script. The base of brahmi script was used to create other scripts.The Brahmi script was used to write Sanskrit or Pali.The inhabitants of Kathmandu valley are in contact with this script and they start to develop their own script known as Prachalit. As time flew away, the Newa people commence designing many beautiful calligraphies. Prachalit, Bhujinmola, and Rañjanā Lipi were broadly used in the middle ages.

Normally, handwritten character recognition has two approaches i.e. Offline Handwritten Recognition [3-4] and Online Handwritten recognition [1-2]. The information from the input scanned images are used in Offline Handwritten recognition. In contrast, online handwritten recognition utilizes the movement of the pen as its input image data. Every person has their own writing styles and even some people write so fast that makes the character so complicated and concise which makes it difficult for character recognition.

In this research, VGG19, ResNet50 and InceptionV3 are implemented to acquire the accuracy rate of ORLHCR with primary Rañjanā Lipi Dataset (RLD) which was collected with the help of Nepal Lipi Guthi teams and other calligraphy enthusiastic. About 6790 precise dataset of isolated character images of 62 classes were developed.

B. Problem statement

Many types of research have been done on recognizing Devanagari script but in the case of Rañjanā Lipi, only limited research found in the case of OCR or digitization of many religious documents, manuscripts, which are in fragile condition and are at risk of deterioration.

Till now, no any dataset were found in digital platform and if someone developed it already, which are not available publicly and handwritten character recognition is not done. So, this research paper address on the development of Rañjanā Lipi Dataset (RLD), and Offline Rañjanā Lipi Handwritten Character Recognition (ORLHCR) with precise result which can aid in digitization and preservation process.

C. Objectives of the study

The main objectives of this research paper is to develop a novel dataset for Rañjanā Lipi and to recognize Rañjanā Lipi Handwritten character using VGG19. ResNet50, and InceptionV3 and compare the accuracy results.

D. Significance of the study

The beneficial of this research paper is to aid for the recognition of Offline Rañjanā Lipi handwritten character with high

ccuracy and also helps other to further research in compound character of Rañjanā Lipi.

E. Limitation

This research paper is only limited to recognize only isolated characters and does not address to compound letter recognition, consonant with identifier, and dynamic segmentation of data.

II. Review of Related Literature

In research, entitled Assamese Character Recognition using (CNN) Convolutional Neural Network, two datasets have been used, which contain a total of 12,863 images. The images consist of 52 Assamese characters, which were collected from a group of more than 200 individuals including students, faulty, and staff from IIT Guwahati in India. The datasets were split into training (70%), validation (20%), and testing (10%) sets. Several architectures such as DenseNet201, ResNet50, LeNet 5, and InceptionV3 were trained and compared. The highest accuracy of 94% was achieved using DenseNet201, in compared to other CNN algorithms [11].

Similarly, in research, entitled "Vietnamese Handwritten Character Recognition using Convolution Neural Network", different classification and algorithms are in use. SVM classification with 39,800 samples achieved an accuracy of 91.3%, while a CNN with 3 convolutional layers and 2 fully connected layers had a 97.2% accuracy rate [12].

In research, named "Manipuri Handwritten Character Recognition by Convolutional Neural Network", 90 different people of age groups and education were involved in the creation of a handwritten dataset. About 4900 sample image datasets are used. Various recognition methods are in use and their accuracy rate varies such as fuzzy features and probabilistic with Artificial Neural Network (ANN) shows 90.3% accuracy rate, binary pattern as vector and NN with back-propagation shows 80%, Binary pattern of pixel density using NN shows 85%, Gabor filter using with SVM show 89.48% accuracy rate, Lenet-5 shows 96.02% accuracy rate, and CNN model shows 98.86% accuracy rate. Among them, the CNN model with 98.86% highest accuracy rate is shown [13].

In a Journal named "Handwritten Devanagari (Marathi) Compound Character Recognition using Seventh Central Moment", the samples with 'Shirorekha', without 'Shirorekha', and the combination of different characters' data is used and the average accuracy rate is 93.87% of some compound characters. The recognition of composite characters was performed using a combination of 7 Invariant Moments and 7th order Central Moments, and a classifier based on SVM was employed. The outcome accuracy rate is 93.87% [16].

In the research paper of "Rañjanā Script Handwritten Character Recognition using CNN", about 17,360 data has been collected from 150 people and later 173,600 data's are generated with data augmentation. The datasets are divided into Training, Testing, and Validation set of 60:20:20 ration and tested using LeNet, AlexNET, ZFNET and a proposed CNN model. The test accuracy rate of 99.73% result achieved by proposed model with 64x64 pixels resolution [25].

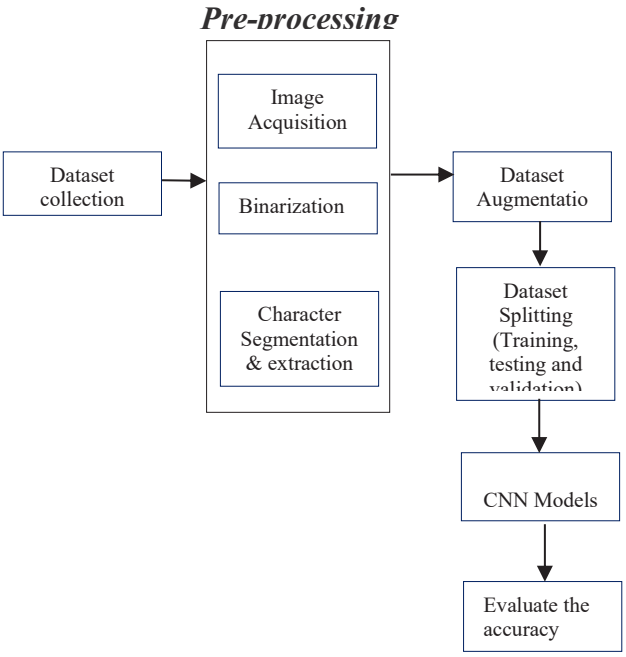

Fig. 1. Methodology

II. Methodology

Rañjanā Lipi dataset needs to develop from the scratch. The dataset is collected with the help of Nepal Lipi Guthi team, calligraphy enthusiasts, learners, and by capturing the image from different places where Rañjanā Lipi is used. This research methodology consists of six parts which are dataset collection or development, pre-processing, dataset augmentation, dataset splitting, fit to CNN models, and evaluate the accuracy rate.

A. Data collection

It's the initial step with lots of hard work that need to be done in the case of Rañjanā Lipi. The isolated handwritten Rañjanā Lipi character will be collected by capturing the images where this script is used. Similarly, can collaborate with the team of Nepal Lipi Guthi, teachers, students, artists, designers, and interested people. Every character data should be of 52-60 different types which have different unique styles. The total number of isolated characters is 62 classes which contains vowels, consonants, and digits. So, about 40 volunteers involve to create a 6792 dataset of 67 x78 pixels.

The filled dataset collected from the volunteers are:

The 62 isolated characters were collected from filled A4 sheet paper.

B. Data pre-processing

Data pre-processing is a significant steps. This steps involve noise filtering techniques. The pixels in the image are classified as either background (colour white) or character (colour black). However, some writers may have mistakenly added extra, unwanted characters or dots in the image, which need to be

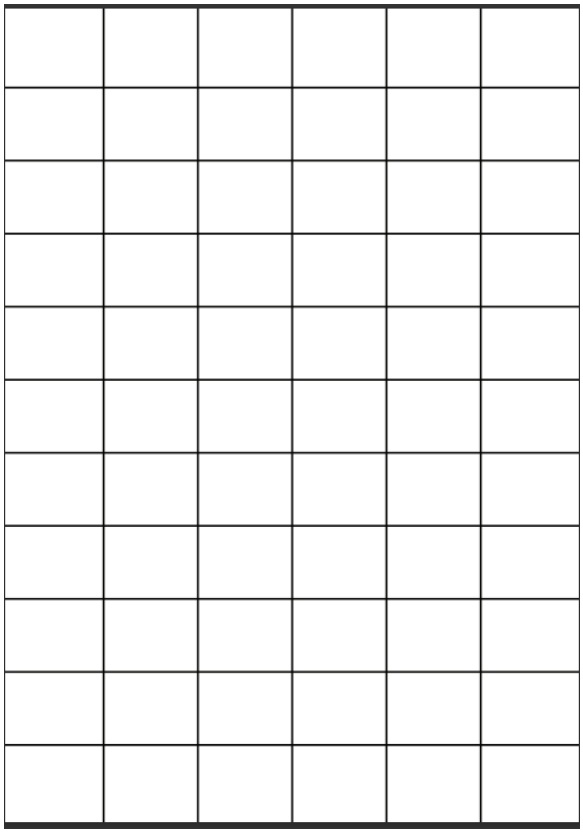

Fig. 2. Sample form to collect the dataset from an A4 sheet.

Fig. 3. Voluntarily collected dataset.

removed through the use of outlier elimination. After this step, image smoothing is performed using the average filter method [11].

In the case of Rañjanā Lipi, the input image is transformed into grayscale and added the Gaussian blur to the converted gray scaled image. A certain threshold is chosen to choose the portion of an image that are relevant to it while ignoring the rest. The threshold value and techniques like THRESH_BINARY_INV AND THRESH_OTSU were used. The morphological transformations required the initial image and the structuring element which decides what kind of action to be performed. The kernel size for MORPH_RECT is 44 x 44 and performs the Dilation, erosion, and open morphological operations. At last, the red rectangle data is segmented or extracted and saved to the segmented datasets directory.

Note: The above explain process is implemented for few dataset to extract the dataset automatically but other dataset are segmented with the help of CamScanner to scan images and convert image to B&W and Figma is used to segment the dataset.

C. Data augmentation

Data augmentation involves using various transformation methods to produce additional data and reduce high variability, leading to the development of improved models [18]. Deep

learning algorithms typically perform best when trained on a large amount of data, but if the available data is limited, the recognition results can be poor. To address this issue, data augmentation techniques are employed to increase the amount of training data and improve recognition accuracy.

Similarly, data augmentation is also used to avoid data over fitting. To generate artificial data into our dataset, rotate a certain degree an image, de-centre it or zoom in or zoom out tiny, and different techniques are used likes vertical/horizontal flips, rescaling, random crops, adjust brightness, rotations, translations, width shift, height shift and many more.

D. Data splitting

Data division is very significant steps and complex part. Destitute preparing and testing of data sets can lead to eccentric impacts on the output of the model which may lead to over-fitting or under-fitting of the data and our model may conclusion up giving one-sided comes about. Normally data are split into 3 sets i.e. train, test, and validation set.

Same as, in this research, data sets are split into 75% train set, 25% test set and from 25% test set data are further divided equally into validation set and test set from test set with random state. To partition data sets into train and test, sklearn model is used. The total number of dataset used in this research is 6,792.

E. Classification and Fit model

In this research, 62 classes (10 digits from 0-9, 16 vowels, and 36 consonants) are classified and the class labels are categorical. Classification is achieved by employing a fully connected layer, where probabilities for each class are obtained from the input data. The input is then assigned to the appropriate class by selecting the class with the highest probability. To predict the model VGG19, ResNet50, and InceptionV3 model is used. Every sample from the training, testing, and validation set is processed to produce a good fit model. The result is the accuracy rate from VGG19, InceptionV3, and ResNet50.

F. Model Evaluation and Verification

In order to determine if a model is producing a good results, it is significant to assess its performance using metrics designed to evaluate the effectiveness of the model. In this research, the

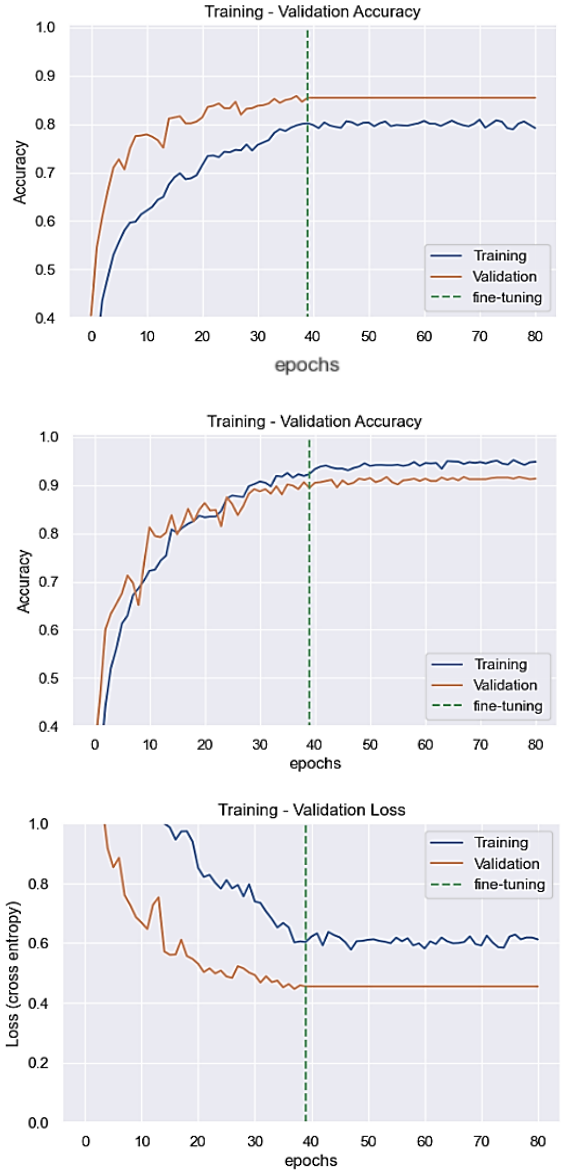

Fig. 4. ResNet50 validation accuracy and loss without tuning learning parameter.

Hold-out method is used where datasets are divided into 75/25 splitting of training, testing, and validation sets and to appraise the model results average precision, recall and F1 score across multiple classes.

III. Results and Discussion

To train the model initially Kaggle is used but due to the limitation of jupyter notebook, later the model was trained in personal PC.

The data are loaded from the local dataset directory with shuffle. The loaded data files are segregate into Target label with their classes. The class are labelled as numerically as [49 42 49 ... 13 37 11] and class label as:

['a', 'aa', 'ah', 'ai', 'am', 'au', 'ba', 'bha', 'ca', 'cha', 'da', 'dda', 'ddha',

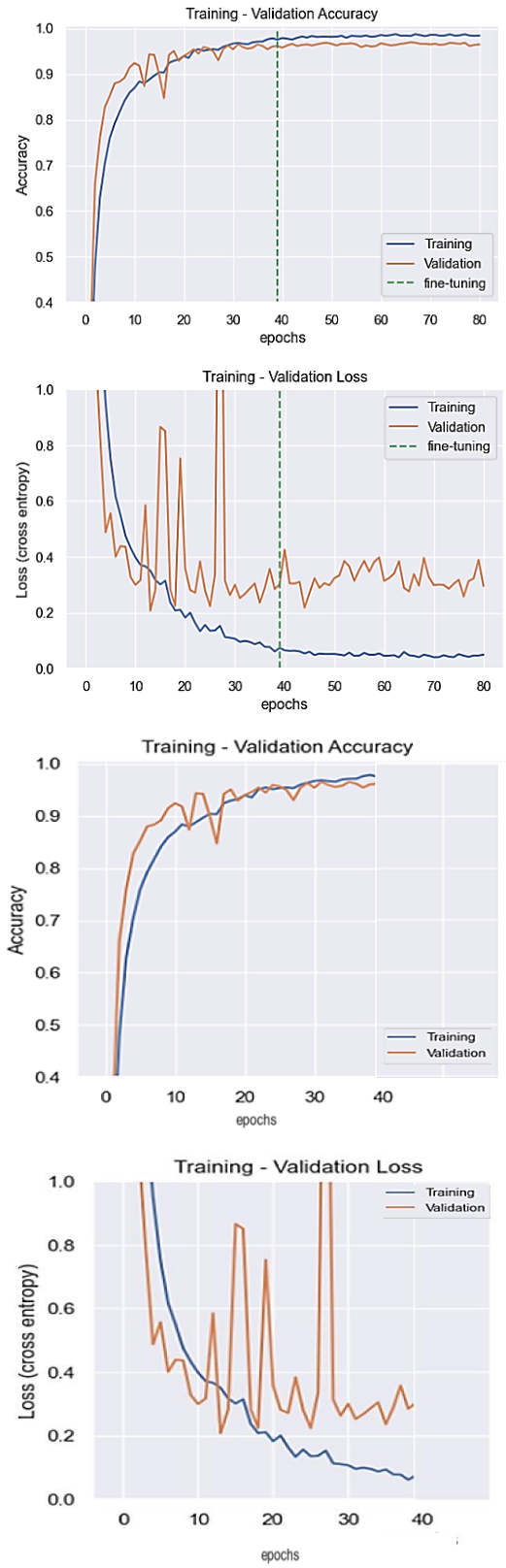

Fig. 5. ResNet50 validation accuracy and loss without tuning learning parameters.

'dha', 'e', 'eight', 'five', 'four', 'ga', 'gha', 'gyan', 'ha', 'i', 'ii', 'ja', 'jha', 'ka', 'kha', 'ksa', 'la', 'lu', 'luu', 'ma', 'na', 'nine,' 'nna', 'nnna', 'nya', 'o',

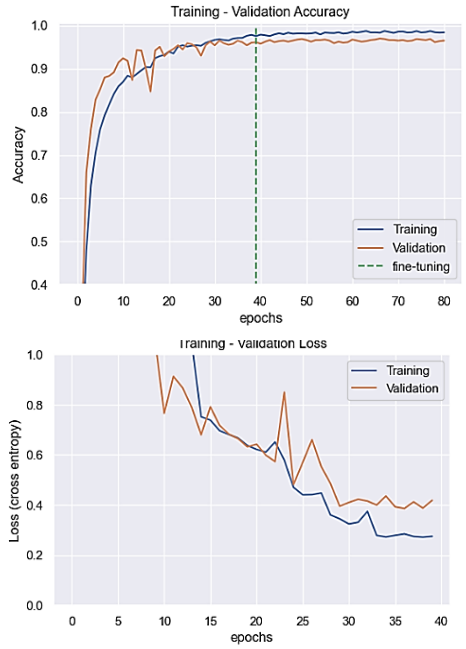

Fig. 6. ResNet50 validation accuracy and loss without tuning learning parameters

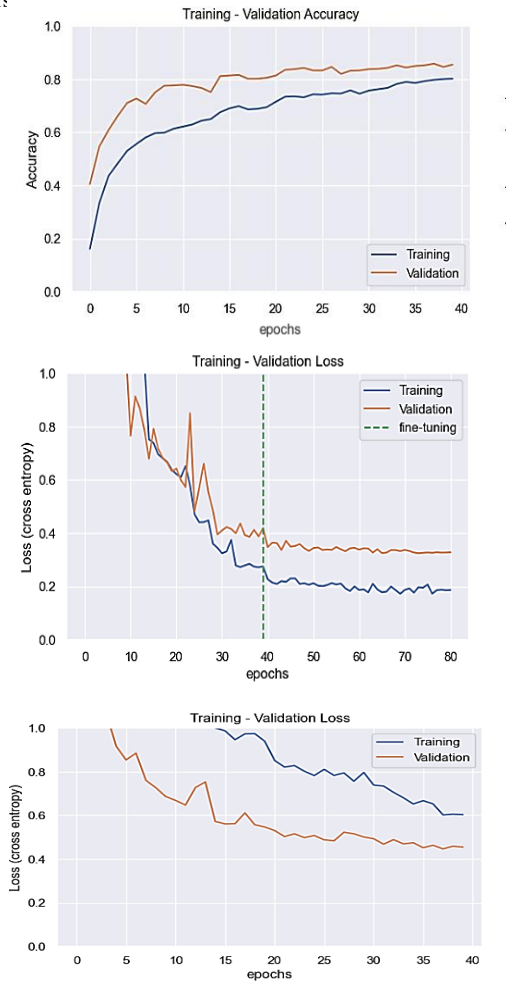

Fig. 7. InceptionV3 training-validation accuracy and loss with and without tuning learning parameters.

TABLE I: VALIDATION ACCURACY AND VALIDATION LOSS OF DIFFERENT ARCHITECTURES

Architecture	Batch Size	Epochs	Accuracy (%)	Loss
ResNet50	32	80	96.82	0.16
VGG19	32	80	92.58	0.29
Inception v3	32	80	85.39	0.43

TABLE II: RESNET50 METRICS WITH ACCURACY, MACRO AVERAGE AND WEIGHTED AVERAGE

Metrics	Precision	Recall	F1 - Score	Support
accuracy			0.97	849
macro avg	0.97	0.96	0.96	849
weighted avg	0.97	0.97	0.97	849

TABLE III: VGG19 METRICS WITH ACCURACY, MACRO AVERAGE AND WEIGHTED AVERAGE

Metrics	Precision	Recall	F1 - Score	Support
accuracy			0.93	849
macro avg	0.91	0.90	0.90	849
weighted avg	0.94	0.93	0.93	849

'one', 'pa', 'pha', 'ra', 'ri', 'rii', 'sa', 'saa', 'seven', 'sha', 'six', 'ta', 'tha', 'three', 'tra', 'tta', 'ttha', 'two', 'u', 'uu', 'wo', 'ya', 'zero']

As we have 62 class which includes vowel, consonants and digits. Every class have their class or labels which is associate with unique value which is also known as character number. The total number of loaded data are 6792. Later the loaded data are further converted into numpy arrays and also convert the image size into 224 x224 images sizes. As if the loaded images contains any varies size then it is convert the images into specific sizes of numpy array which helps to normalize the variability.

The experimental result of Vgg19, ResNet50, and InceptionV3 with 40 epochs without hyper parameter tuning and with tuning learning parameter 80 epochs with accuracy and loss of Training-Validation are calculated. The below graph shows the relationship between the training and validation with Accuracy and Loss.

From figure 4, 5, 6, and 7, by tuning the hyper parameter learning rate from 0.001 to 0.001, the performance of the model is improved in performance on ResNet50, VGG19, and InceptinV3. The tune epochs is the total epochs from the summation of previous epochs and tune hyper parameter learning epochs. The vertical dotted green line represents with and without tuning learning hyper parameter.

By fine-tuning the learning parameters, the deep learning models, ResNet50, VGG19, and Inception v3 trained on a specific Rañjanā Lipi Dataset with 80 epochs and a batch size of 32. An epoch is one complete iteration through the dataset, and the batch size refers to the number of samples processed and pass through the network. When the data is tested from Rañjanā Lipi Dataset, 96.82% accuracy rate is achieved by using ResNet50. Similarly, 92.58 % by using VGG19, and 85.39% by using Inception v3. These accuracy rates are specific to the Rañjanā Lipi Dataset, training conditions, and evaluation criteria and may note generalize to other datasets or use cases.

Fig. 8. Prediction of Rañjanā Lipi image by using ResNet50

TABLE IV: INCEPTIONV3 METRICS WITH ACCURACY, MACRO AVERAGE AND WEIGHTED AVERAGE

Metrics	Precision	Recall	F1 - Score	Support
accuracy			0.85	849
macro avg	0.84	0.84	0.83	849
weighted avg	0.87	0.85	0.85	849

The Rañjanā Lipi dataset of every class does not contains equal set of data. So, the weighted average metrics is used. Table II-IV show the difference in architecture performance metrics. The combined accuracy is calculated from the custom Rañjanā Lipi Dataset. The results with accuracy and loss with tuning the learning parameters are shown in table below:

Tables II, III, and IV show the performance metrics commonly used to evaluate the classification tasks of ResetNet50, VGG19, and InceptionV3. The accuracy metric is used to measure the percentage of correct predictions made by a model. Similarly, Macro average is a metric that is used to calculate the average of metric scores for each class, treating each class as equally important, and weighted average is a metric

In Table II, the accuracy rate for ResNet50 show the 0.97 which means that the model has a 97% accuracy in its positive class predictions, recall of 0.96 means that the model is able to detect 96% of the positive class samples, and F1 score of 0.96 means that the model has a good balance between precision and recall, with a score close to 1.

Similarly, in Table III, the accuracy rate for VGG19 show the 0.93 which means that the model has a 93% accuracy in its positive class predictions, recall of 0.90 means that the model is able to detect 90% of the positive class samples, and F1 score of 0.83 means that the model has a good balance between precision and recall, with a score close to 1, and in Table IV, the accuracy rate for InceptionV3 show the 0.85 which means that the model has a 85% accuracy in its positive class predictions, recall of 0.84 means that the model is able to detect 84% of the positive class samples, and F1 score of 0.83 means that the has not a good balance between precision and recall for and its score is not above 0.9 and close to 1. At last, the model has successfully recognized the text from the images. The sample recognized data from the sample data and mobile application are shown in Fig. 9.

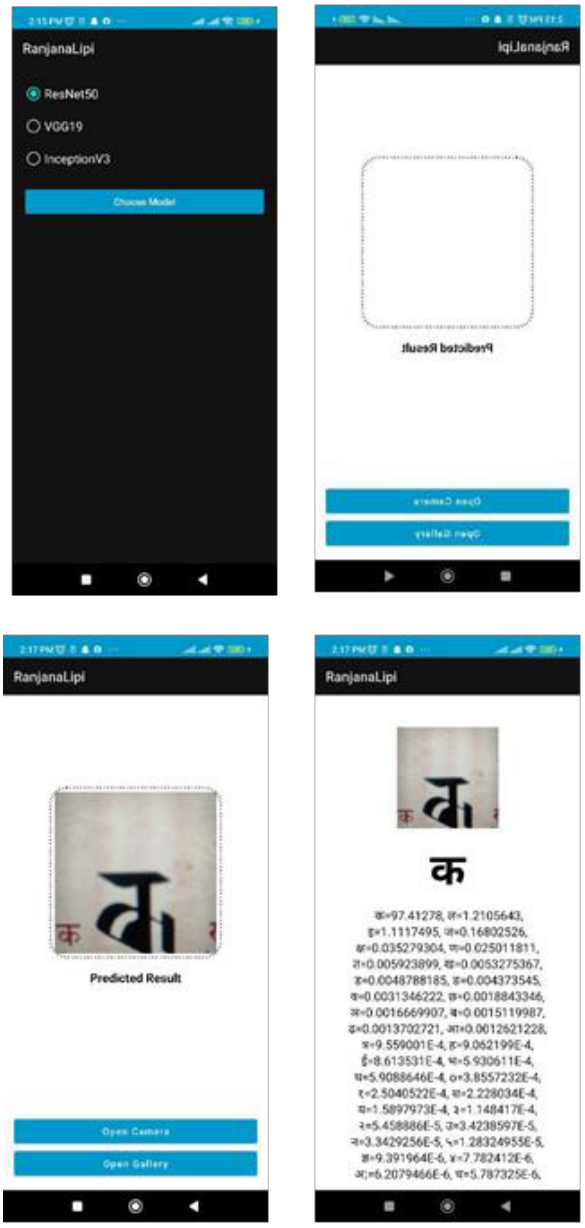

Fig. 9. Prediction with the ResNet50 model using a custom-made Android application.

The trained model is further converted into.tflite to integrate the model in android application and to implement OCR of Rañjanā Lipi Character offline.

V. CONCLUSION

In this research, the best performing model is ResNet 50 which achieved the accuracy rate of 96.82% by improving the learning parameters. Comparing Precision, recall, and F1 score of ResNet50, VGG19, and InceptionV3, and find that ResNet50 has scores closer to 1 than VGG19 and InceptionV3, this suggest that ResNet50 is a more accurate model than VGG19 and InceptionV3. One important note that the performance of a model can vary depending on the specific dataset and task being evaluated.

REFERENCES

[1] S. España-Boquera, M.J. Castro-Bleda, J. Gorbe-Moya, F. Zamora-Martinez, “Improving Offline Handwritten Text Recognition with Hybrid HMM/ANN Models”, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 33, no. 4, pp. 767 - 779, 2011.

[2] Yanwei Wang; Xiaoqing Ding; Changsong Liu, “Topic Language Model Adaption for Recognition of Homologous Offline Handwritten Chinese Text Image”, IEEE Signal Processing Letters, vol. 21, no.5, pp. 550 - 553, 2014.

[3] Daniel Keysers, Thomas Deselaers, Henry A. Rowley, Li-Lun Wang, Victor Carbune, “Multi-Language Online Handwriting Recognition”, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 39, no. 6, pp. 1180 - 1194, 2017.

[4] Zecheng Xie, Zenghui Sun, Lianwen Jin, Hao Ni, Terry Lyons, “Learning Spatial-Semantic Context with Fully Convolutional Recurrent Network for Online Handwritten Chinese Text Recognition”, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 40, no. 8, pp. 1903 - 1917, 2018.

[5] Cheng-Lin Liu, Fei Yin, Qiu-Feng Wang, Da-Han Wang, “ICDAR 2011 Chinese Handwriting Recognition Competition”, 2011 International Conference on Document Analysis and Recognition, 2011.

[6] Michael Murdock, Jack Reese, Shawn Reid, "ICFHR2016 Competition on Local Attribute Detection for Handwriting Recognition", 2016 15th International Conference on Frontiers in Handwriting Recognition (ICFHR), 2016

[7] Jangid Mahesh and Sumit Srivastava, "Handwritten Devanagari Character Recognition Using Layer-Wise Training of Deep Convolutional Neural Network," Journal of Imaging, February 2018.

[8] H. Singh, R.K. Sharma and V.P Singh,” Online handwriting recognition systems for Indic and non-Indic scripts: a review” Artif Intell Rev 54, 1525–1579, 2021.

[9] Manoj Sonkusare, Gupta Roopam, and Asmita Moghe, "Palm-Leaf Manuscript Character Recognition and Classification Using Convolutional Neural Networks," January 2019.

[10] Adith Narayan, Raja Muthalagu, “Image Character Recognition using Convolutional Neural Networks”, June 2021

[11] Mihir Yadav, Divyansh Mangal, Srinivasan Natesan, Marcin Paprzycki, and maria Ganzha, “Assamese Character Recognition using Convolutional Neural Networks”, 2021

[12] Truong Quang Vinh, Le Hoai Duy, Nguyen Thanh Nhan, “Vietnamese handwritten character recognition using convolutional neural network”, April 2020

[13] Sanasam Inunganbi, Prakash Choudhary, and Khumanthem Manglem, “Manipuri Handwritten Character Recognition by Convolutional Neural Network”, pp, 307-318, 2020

[14] Saniya Ansari, Bhavani S, and Udaysingh Sutar, "A Novel approach towards Online Devanagari Handwrtten word recognition based on robust feature extraction method and FFNN classifier," International Journal of Advanced Research(IJAR), vol. 5, no. 8, pp. 764-776, 2017.

[15] Mohammed N AlJarrah, Mo’ath M Zyout, Rehab Duwairi, “Arabic Handwritten Characters Recognition Using Convolutional Neural Network”, 2021

[16] P. E. Ajmire, R. V. Dharaskar, and V. M. Thakare, "Handwritten Devanagari (Marathi) Compound Character Recognition using Seventh Central Moment," International Journal of Innovative Research in Computer and Communication Engineering, vol. 3, no. 6, June 2015.

[17] Kaggle, Competition Notebook, Kaggle, https://www.kaggle.com/cdeotte/25-million-images-0-99757-mnist

[18] C. Shorten and T. M. Khoshgoftaar, “A survey on Image Data Augmentation for Deep Learning,” J. Big Data, vol. 6, no. 1, pp. 1– 48, Dec. 2019, doi: 10.1186/s40537-019-0197-0

[19] Mohammed N AJarrah, Mo’ath M Zyout, Rehab Duwairi, “Arabic Handwritten characters recognition Using Convolutional Neural Network”, 978-1-6654-3351-8/21/$31.00 ©2021 IEEE

[20] Kavitha B.R, Srimathi C, “Benchmarking on offline Handwritten Tamil Character Recognition using convolutional neural networks”, 7 June 2019

[21] Hassan El Bahi, Zouhir Mahani and Abdelkarim Zatni, “An offline handwritten character recognition system for image obtained by camera phone”, 15 October 2015

[22] I Khandokar, Md M Hasan, F Ernawan, Md S Islam, M N Kabir, “Handwritten character recognition using convolutional neural network”, 2021

[23] Gu J, Wang Z, Kuen J, Ma L, Shahroudy A, Shuai B, Liu T, Wang X, Wang G, Cai J and Chen T 2018 Patt. Recognition 77 354

[24] Y. Lecun, L. Bottou, Y. Bengio and P. Haffner, ”Gradient-based learning applied to document recognition,” in Proceedings of the IEEE, vol. 86, no. 11, pp. 2278-2324, Nov. 1998; DOI: 10.1109/5.726791,

[25] Jen Bati, “Rañjanā Script Handwritten Character Recognition Using CNN”,2022