Creation of Mnemonics for Hindi alphabets using CNN and Autoencoders

MSc Research Project
Data Analytics

Palak
Student ID: 18185461

School of Computing
National College of Ireland

Supervisor: Dr. Vladimir Milosavljevic

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Palak		
Student ID:	18185461		
Programme:	MSc in Data Analytics	**Year:**	2019-2020
Module:	MSc Research Project		
Supervisor:	Dr. Vladimir Milosavljevic		
Submission Due Date:	28th September 2020		
Project Title:	Creation of Mnemonics for Hindi alphabets using CNN and Autoencoders		
Word Count:	9566 (Including references) **Page Count:** 23		

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	
Date:	25th September 2020

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Creation of Mnemonics for Hindi alphabets using CNN and Autoencoders

Palak
18185461

Abstract

Mnemonic helps the brain in retaining memory via visual, audio, textual or any other means. The use of Mnemonics is a comparably lesser explored method for language learning, even though it is fairly effective. The research generates visual mnemonics for the Hindi language using machine learning algorithms to make Hindi character learning stimulating for learners. The creation of mnemonics is a tiresome process; hence this research enabled the algorithms to create visual mnemonics for learners instead. The research used Convolutional Neural Network (CNN) for classification of handwritten Hindi characters and Autoencoders for feature extraction of characters as well as potential mnemonic images. The entire research is divided into four related stages, each with its own objectives. CNN gave an accuracy of 98.48% and autoencoder had MSE score of 0.038. The images generated by the autoencoder weren't entirely visible for normal eyes, hence they were evaluated using Euclidean distance with the help of nearest neighbours algorithm. The resultant images were suggestions that could work as mnemonics; however, it depends on the individual to validate the impact of any of the suggested images.

1 Introduction

The coming age technology has unleashed another realm into the universe, i.e. the virtual realm (Lundin, 2019). Electronic learning exists in this realm which has enabled a significant shift for the educators and the learners. E-learning is the future and thus, it deserves all the enhancements it could get. This is why E-learning is the base domain of this research. This research focuses on promoting the learning of languages virtually. E-learning is also responsible for helping in the imperative development of the brain. This area has been ever improving since years now and it doesn't seem to stop. If anything, E-learning is deepening it roots with the assistance of emerging technologies like Artificial Intelligence, Virtual Reality, Augmented Reality among others (Gunasekaran, McNeil and Shaul, 2002).

As mentioned above, this research explores the learning of languages via electronic means. The language chosen for this purpose is Hindi. Hindi is one of the ancient languages which is hugely regarded in India and its adjoining neighbours (Kimmel, 2020). Approximately 490 million [1] of world's population is acquainted with Hindi. It dominates the remaining 22 languages existent in India. Hindi, therefore, appeared to be an appropriate choice for this

[1] Source URL: https://www.vistawide.com/languages/top_30_languages.htm

research. In order to learn any language, the learner requires to start with the very basics, i.e., the characters of the language. This research focuses on initiating a learning process for the enthusiasts. Hindi script has about 36 characters and 10 digits. Even for the native learners, this language creates challenges because of its trivial structures. Hence, a learning aid could prove to be extremely useful.

Mnemonics is the most crucial aspect of this research. This progression of E-learning for the Hindi characters is heavily assisted by Mnemonics. Anything that helps to retain a memory of something is Mnemonic (Rohland, 2019). There are various kinds of Mnemonics, namely textual, audio, visuals and so on. Knowingly or unknowingly, each of our brains has implemented Mnemonics in daily life. For instance, V.I.B.G.Y.O.R. is a textual mnemonic for the colours of the rainbow in the correct order. The Medieval Era is not known for its literacy, yet there have been proofs of the usage of various symbols and pictures during that time. Even parents attempt to teach language to their kids with some visual of audio aid. Therefore, the amalgamation of Mnemonics in the research for ministering the e-learning process of Hindi would definitely prove to be beneficial.

In the area of data analytics and machine learning, there have been a few works (Tamara, Rusli and Hansun, 2019) (Ying, Rawendy and Arifin, 2016) who have integrated Mnemonics into language learning in the past. However, these researches utilized the machine learning algorithms to evaluate the findings rather than utilizing them to obtain the findings. This research depended on the algorithms for the entire learning process. This research evaluated the handwritten Hindi characters and enabled the algorithms to create Mnemonics, unlike the existing state of art. This research, hereby, boosts the participation of data analytics in the domain of e-learning. It proves that the machine learning algorithms have more potential than they are given credit for.

This research was purposed to enable the machine learning algorithms create Mnemonics for the Hindi characters. The creation of visual mnemonics is a task that requires human intellect and creativity along with a huge amount of efforts. The entire procedure of creating Mnemonics can be tiresome. The conventional process is initiated by studying the language character for which the mnemonic is needed to be created. Upon understanding the structure of the character, an entity or object needs to be thought about to map it with the character. For instance, a close Mnemonic for the English alphabet 'A' could be the Eiffel Tower because of the resemblance between the two. This results in creating a significant impact on the learners' mind while recalling a certain character. This entire thought process and manual labour could easily be avoided if the machine learning algorithms are utilized for the same. The research used Convolutional Neural Network (CNN) and Autoencoders to achieve the Mnemonics for the characters of Hindi script, also known as the Devanagari script.

The research is initiated by classifying handwritten Devanagari/Hindi script characters and identifying it. The terms Hindi and Devanagari are used interchangeably in the paper. Further, an autoencoder is trained to extract essential features from the handwritten character dataset and reconstruct the characters. Based on the appropriate parameters recognized via this autoencoder, another autoencoder is trained to carry out the same task for Mnemonic dataset. The latter autoencoder consumes a Hindi font dataset and maps one of the trained Mnemonic images for each character of the language, including numeric digits.

1.1 Research Question

The question addressed by this research is – How efficiently can the machine learning algorithms refabricate the process of creating Mnemonics for Hindi language?

The prime objective of this research is to encourage learning via Mnemonics. The language learning process speeds up with the use of Mnemonics. There are some other objectives as well that are fulfilled by this research. They are as follows: -

1. The research also targets ease the manual work in the laborious process of creation of Mnemonics.
2. Through the medium of this research, it is also desired to increase the involvement of data analytics and machine learning in the domain of e-learning.
3. It proves that the creation of Mnemonics for the characters of a language is possible without high human efforts, creativity and time.

1.2 Novelty

This research is intriguing because it holds a level of novelty in the various areas it covers like e-learning, data analytics and mnemonic learning.

1. The research reflects novelty because as far as it was explored and observed, there isn't a research that has used Mnemonics to teach the Hindi language. There have been a few research that have made character learning easy via mnemonical learning for languages like Japanese (Tamara, Rusli and Hansun, 2019), Chinese (Ying, Rawendy and Arifin, 2016), etc.

2. The mnemonics were created manually by the researchers for each of the characters. This research relied upon autoencoders to generate mnemonics. This not only saves up time and efforts, it might also prove to be more effective than the manual creation of mnemonics.

As the sections progresses, more aspects of this research would be explained. Section 2 of the report includes the review of existing state of art. Section 3 simultaneously talks about the stages in Research methodology, the results and the evaluation. Section 4 contains the Discussion, followed by Conclusion and Future work in the Section 5.

2 State of Art

This research of Mnemonics for Hindi characters covers certain intriguing domains that aren't entirely related to each other directly. This research creates an amalgamation of these in order to achieve the target. The literature review is, therefore, outlined in various subsections to explore the existing works in all these areas. While no literature entirely relates to this research, there are some research that gave insights on the methods and technologies. This research extensively enhances and extends works based on Mnemonics.

2.1 Hindi: An upcoming promising choice for language study

Hindi holds another level of respect as a language having its roots from the ancient language, Sanskrit (Kimmel, 2020). It is, therefore, a language to divulge into and explore. Time and now, researchers have been analysing Hindi in various domains. The authors Jain, Darbari and Bhavsar (2017) highlights the usage of ATV, expanded as Automatic Text Visualization, for providing assistance in cognitive terms to the individuals who face challenges in learning. There is a compelling comparison of plain and visual text in relation to the Hindi script, also called Devanagari. The ATV based system enabled the generation of examples to illustrate the nouns, verbs, adverbs, etc. of Hindi language and compared the visuals to the text examples. As the left and right cerebrum hypothesis says, right side is more capable of visualizing while reading text but the left side is short of creative abilities. The left side has an upper hand in handling quantitative dealings. The reading disorder, Dyslexia disables the individual to identify the characters which makes it hard for them to learn. The created system, Preksha, enables the dyslexic individuals of a comfortable environment for learning purposes. Likewise, the idea behind this research is to utilize the right side of the brain to teach language learning using Mnemonics. Mnemonics make the learning more creative and intriguing. Mnemonics would create an impression of the structure of the Hindi character so it becomes easy for the individual to recall it. The individuals with Dyslexia have a lesser active left side of the brain, making their right sides more dominant. Hence, Mnemonics would be easier to grasp because of the attractive images, thus visual mnemonics were preferred for this research over textual mnemonics. This research, however, is equally encouraging for both non-dyslexic and dyslexic language enthusiasts. The mapping of natural images to Hindi characters simplifies the understanding of the shape and structure of a particular Hindi character.

Hindi script, or Devanagari being an ancient language, a number of manuscripts have been created in it. Those documents have high historical significance because the printing technologies were discovered late in the upcoming eras. The characters and/or the script is similar to a lot of other languages, including Nepali, Marathi, Hindi among others, across different states and even countries. The ancient documents are, thus extremely sensitive and fragile in nature because of which they weren’t made accessible for research purposes. It was with the digital age that the documents could be analysed. The research of authors Narang, Jindal and Kumar (2019) attempted to recognize the characters of the ancient documents that belonged to different museums and libraries. The digitization of these libraries enabled them more accessible to the researchers. Narang, Jindal and Kumar (2019) used different classifying techniques like CNN, MLP, Neural Network, Random Forest and RBF-SVM to digitize the documents conveniently. Optical Character Reader (OCR) technology was also used for the task of identification of more than 6000 characters. The feature like horizontal and vertical peak extents, open and intersection endpoints and centroid were used for recognition of vowels and consonants in Devanagari script. This research attempts to recognize the Hindi characters as well and accolades the attempt by the authors of research Narang, Jindal and Kumar (2019) for conserving ancient documents despite of the possible challenges like ink stains and blurriness. The structural insights of the Devanagari characters proved to be immensely helpful for this research.

Along with the alphabets of the Devanagari script, this research also generated appropriate Mnemonics for Devanagari numeral digits too. The paper by Reddy, Rao and Raju (2018) trained a profoundly effective model for recognition of Hindi numerals by obtaining 99% accuracy. The numeral system for Hindi is same as the English numeral system of base 10. CNN was utilised by the research along with an optimizer, namely RMSProp which is yet to be a published algorithm. It is considerably challenging to get out of flow numeral digits back as numbers without human involvement of any kind. It is difficult to digitize large set of characters. Therefore, the same concept of joined numerals and alphabets is used by Captcha, along with some random noise for distinguishing actual users and bots. The paper provided great insights about the importance of optimizers on the research. Another research by Zhan *et al.* (2019) pondered more on the recognition of numeric digits rather than the alphabets of Devanagari. The use case for the paper was the Indian Postal System that works on the Indian Postal Index numbers. The system involves the usage of a 6-digit unique code for identification of local or interstate numbers. A novel architecture of CNN along with CTC has been utilized in the mentioned paper without an RNN network. An understanding obtained from this and the previous research was that both made use of CNN for character recognition for their respective purposes. Also, it could be observed that CNN is more efficient when used in combination with other techniques, hence, this research uses autoencoders to compliment CNN.

In the past, researchers have also attempted to perform sentiment analysis in Hindi language. The paper by Yadav and Bhojane (2019) gathered user reviews from various areas in Unicode 8 to train a semi-supervised model. The model classified the polarity of 700 documents in Hindi based on the opinions present in the user reviews. There were 3 different approaches established to achieve the same, one of the approaches being Sentiment WordNet. It made use of more than 23k words to classify the polarity more effectively by reducing the noise. This research, however, is more character unlike the latter paper.

2.2 Mnemonics: A memory retention technology

Mnemonics have proven to be useful directly or indirectly in numerous domains. It serves the tasks of memory retention very efficiently. Similar to this research, one of the domains in which Mnemonics have proven their worth is language learning.

Through the years, there have been various attempts by different researchers to incorporate the usage of Mnemonics to simplify the process of learning a language or its characters. This research was aimed to achieve the same goal. Another such paper by Tamara, Rusli and Hansun (2019) encouraged language learning using Mnemonics. The mnemonics were created for Japanese language and an e-learning application was created for both iOS as well as Android users. There were groups of students formed to analyse the before-mnemonic and after-mnemonic results. CNN was utilized to evaluate the input of the Japanese character given by the students after studying the Mnemonic for that character. The students expressed a significant desire towards the Mnemonical approach. This research extends the research of Tamara, Rusli and Hansun (2019) as the mnemonics created in the latter were manual. The entire purpose of this research is to eliminate that laborious task. Also, the paper utilized machine learning algorithm (CNN) just for the evaluation of their research whereas this research trained algorithms (CNN and Autoencoders) to actually create Mnemonics. It resulted in utilizing the full potential of the machine learning techniques.

Apart from Japanese, Chinese is another language that was attempted to be taught using Mnemonics. The study by Ying, Rawendy and Arifin (2016) created a gaming education platform for learning Chinese. Chinese characters could be difficult to understand for toddlers, hence, a game for mobile was developed to accomplish easy learning for the language characters. A blueprint is proposed in the study for creation of an application or a game. However, a huge gap in this study could be found in the lack of any machine learning algorithm. As this research and many others have explored, machine learning techniques are capable of easing the learning process to a whole other level. Therefore, this research incorporated CNN as well as Autoencoders to attain the target. This research also enables the machine learning algorithms to create Mnemonics instead of humans.

Along with character memorization, mnemonical learning is beneficial in improving vocabulary as well. The paper by Leelanupab and Anonthanasap (2017) depicts the implementation of mnemonics for boosting the vocabulary for Japanese language. The paper uses a phonetic algorithm to generate mnemonic keywords via a system named JemSoundex. Upon evaluation and analysis using various retention tasks, the system demonstrated performance advances among the learners. However, this research would be focused on visual mnemonics rather than textual mnemonics to enable language learning. Visual Mnemonics might prove to be more effective than the textual ones since they leave an impact on the brain of the learner. Vocabulary isn’t just limited to languages. The study by Bahrami, Izadpanah and Bijani (2019) illustrates the use of Mnemonic for musical vocabulary. There was a division of students into two groups, namely experimental group and control group. Following that, a 14 words thread was synced to a song and given to the groups. Subsequently, it was found that the performance of experimental group was significantly higher than the performance of control group. This collaboration of learning and music to form Mnemonics is an innovative way to encourage young people. The visual mnemonics generated by this research would also be an interesting method of learning for the enthusiasts.

The applications of Mnemonics aren’t just limited to languages and vocabulary. It is beneficial for easy learning and memorizing of other subjects as well. It is very well established by the author Mahaffey (2020) that chemical mnemonics can also prove to be useful for children studying Chemistry in school. The students had a positive response towards the mnemonic system and memorizing Chemistry elements of the course. Geography (Haydon, Musti-Rao and Alter, 2017) isn’t an unexplored domain in terms of Mnemonics either. A group of four students who were mildly to moderately disabled were chosen for the study of choral responses.

During the Geography lectures, the results were recorded for both choral responses as well as choral responses with Mnemonics. The latter presented improved results over the former one. Hence, the value of Mnemonics has been recognized in various fields and the results are rarely disappointing.

It is understood from the mentioned literature that Mnemonics have a substantial role in easing and simplifying different learning processes. However, it could be surprising how useful Mnemonics are in other areas as well. Mnemonics have been incorporated to assist in matters related to security (Song *et al.*, 2019). Generating passwords via the proposed system Alphapwd would avoid any unknown attacks and upgrade the security. The proposed technique combined various keyboard strokes and their writing order to generate passwords. A number of passwords could be simulated on keyboard for each of the alphabets. This makes it easy for the user to memorize their passwords while retaining the security of the user.

While analysing the existing literature, it was observed that Mnemonics are reliable as a method of learning and memorizing. Hence, this research chose Mnemonics upon looking at capabilities and competence level of learning it possess.

2.3 CNN: A benchmark for handwriting recognition

Convolutional Neural Networks or CNN are increasingly being used in recognising textual images such as printed and handwritten character images. The literature by Neri *et al.* (2020) supports the aforementioned statement. The paper discusses a CNN model that has been used to classify two datasets which are able to tackle image classification problems. Despite the fact that CNN can perform quite well without pre-processing of the original images, they spent much efforts at pre-processing the image datasets and obtained 98% accuracy. This led to the conclusion that pre-processing the data can, as a matter of fact, provide better result and performance to some degree for a CNN model. This research also gained beneficial knowledge about the importance of training and pre-processing of data for better results. Another research by Wen, Shao and Zheng (2019) compared transitional neural networks with CNN and DCNN (Deep Convolutional Neural Networks) for identifying handwritten characters, particularly digits. The study yielded 99% accuracy which is better than the previous research Neri *et al.* (2020). CNN results were better than other types of models comparatively and combined with DCNN based Alex Network or AlexNet, a considerable performance and accuracy improvements were noticed. Conclusively, it proved to be beneficial for the research because DCNN combined with AlexNet is proven to be effective.

Another study by Alani (2017) follows the same idea as this research that CNN is effective in recognising digits and characters in various languages. They used digital input devices to scan and identify the given input using a trained CNN model. The authors had RBM or Restricted Boltzmann Machine (RBM) added to the existing model which led to some improvements over the existing methods. Restricted Boltzmann Machine was their choice because it allows extraction of individual features based on their value from raw data and thus higher value features can be separated. The study was used to identify Arabic numeric characters using the feature extraction phase and obtained an accuracy of more than 98% which is less when compared to other research of Wen, Shao and Zheng (2019) because some key points were lacking. Nonetheless, the idea of feature extraction was useful for this research as well. The technique of extracting features is autoencoder for this research which differs from the method (RMB) used in the previously mentioned study. The research by Sufian *et al.* (2020) yet again converges back to CNN and digit recognition but for a different Indian language called Bengali. A model called BSDNet was built to identify Bengali numerals, purely based on densely connected CNN. In the earlier state of art methods, it was observed that some form of pre-processing steps was required for data which is based on augmented methods as well as untraditional approach methods which was also the case for this research. This method was better in terms of accuracy and obtained about 99.78%. To conclude the results of the research discussed above, it can be said that the accuracy dependency on the model type used along with number of layers/nodes is dependent on the data pre-processing as well as the feature extraction stages of the process. Our research would also rely on these findings and will focus on pre-processing for better results and accuracy.

While a lot of work has been done on the handwriting and digit identification, there are many other studies in the past that didn't involve handwritten character or digit recognition. Starting with study by Kalbhor and Deshpande (2018) where digits were represented by signs and gestures instead of handwritten characters. The study was not limited to programs but involved an Internet of Things (IoT) kit along with several sensors fitted into a glove. The hardware is proposed for measuring things like rotation and scale but not used. The model itself was trained and tested using contour-SVM and CNN while making changes to parameters like scale, rotation and orientation. CNN outperformed the contour by a very high margin. CNN got an accuracy of 98.31% whereas contour-SVM was limited to only 69%. The study was better suited for entirely different audiences like the ones with hearing or speech disabilities. This research, on other hand, aims to help people who would like to learn Hindi alphabets visually and hereby, doesn't target any specific kind of audience.

Neural network, in general including CNN, is quite popular and an increasing number of further studies are conducted to aid these methodologies. The world now has better performing models which was earlier not possible via traditional learning models, examples being study by Arif *et al.* (2018) and Siddique, Sakib and Siddique (2019). The authors changed various factors while diving deep into CNNs and studied differences in accuracy after each tweak that they made into the number of hidden layers in the model. The latter analysed over 15 different epoch cycles by modifying the popular dataset called MNIST or Modified National Institute of Standards and Technology dataset. The other study by Arif *et al.* (2018) didn't have any kinds of limits on epochs. Study by Arif *et al.* (2018) illustrated the loss whereas study by Siddique, Sakib and Siddique (2019) discussed the accuracy after these modifications. This research gathered a lot of insights about parameter tuning for CNN from these literatures.

2.4 Autoencoders: A feature extraction and reconstruction tool

CNN, as effective as it is for recognizing and classifying handwritten characters, it couldn't aid in generating Mnemonics for the research. Therefore, autoencoders were included as an integral part of the research. There have been studies (D'Angelo, Ficco and Palmieri, 2020) (Fu *et al.*, 2019) that have explored the applications of Autoencoders for detection. The paper by D'Angelo, Ficco and Palmieri (2020) addresses the possibility of autoencoder for detecting malwares in Android. Android especially was chosen because it is comparatively more open and popular among all the other cellular operating systems. One of the key aspects highlighted in this paper was the decision of choosing the right feature for training the machine learning algorithm. API calls from the cell phones were also made to achieve the research target. Auto encoders were used for extracting essential features from the sparse matrices that were obtained from the API calls. Any type of malware present in the Android was identified when the aforementioned system was fed to an artificial neural network for classification. Not just malware, auto encoders could be utilised fall detecting any kind of speech deception. The study attempted to achieve the same using a noise autoencoder. The semi supervised additive model would overcome the issue prevalent in algorithms that were used earlier, the issue being insufficient data that is also labelled. The paper highlighted that the implemented model performed better than the other existing models. It was able to achieve that even with reduced amount of labelled data. The aforementioned research utilised autoencoders to focus on detection of negative elements to safeguard themselves and their systems. However, this research is more bent towards detecting similarities between two images by extracting their significant features. Autoencoders have distributed uses and this research targeted to utilize it to their more potential.

One of the prime purposes of autoencoders is to reduce dimensions and extract essential features. A number of studies have been done to apply the autoencoders for the same purpose. The paper by Kumar and Nair (2019) comprised of a coupled training approach for convolutional autoencoder. Autoencoders like every other deep learning technique requires the optimal tuning of the parameters. This ensures smooth training of models with less memory usage and less amount of training time. The importance of tuning parameters is highlighted since it is capable of affecting the accuracy of the model, directly or indirectly. This research put a significant amount of efforts for deciding upon the parameters like number of layers or number of neurons in the architecture of autoencoders.

The decision of choosing the correct type of auto encoder for the research was a trivial and time-consuming task. A number of existing works were studied to understand the implementation of the different types of autoencoders. The paper by Wang *et al.* (2019) attempts to overcome the shortcomings of supervised learning by integrating sparse autoencoders (SAE) in it. The SAE enabled extraction of the defective features in order to diagnose any possible faults in the air conditioning systems. Support Vector Machine (SVM) was another method used by the research to compare the results with SAE. As expected, improved results of accuracy and recall along with precision were obtained from the SAE system. It was also found out that the data size and efficiency were directly related to each other. It might or might not be a useful relation with respect to the physical machine's configurations and abilities. Adversarial Variational Autoencoders Lin, Mukherjee and Kannan (2020) is another type of autoencoder that has been used to sequentially analyse single celled RNA. The study proposes an implementation of Adversarial Variational Autoencoders as a dimensionality reduction tool. The system resulted giving better outputs than the clustering algorithm used earlier. Hence, Adversarial Variational Autoencoders helped in reducing the complications faced while sequential analysis of RNA cells. It was observed in both the studies that autoencoders stepped up the results of the conventional systems and gave valuable insights for this research. Since this research is entirely dealing with images, convolutional autoencoders were adopted for 2 stages of this research.

Autoencoders are not limited to a single purpose and there are many areas where they are often used like already shown above. This research is specific to e-learning and autoencoders are often used here as well. The next study by Almotiri, Elleithy and Elleithy (2017) is another one of such examples where autoencoders are used in e-learning. It demonstrates autoencoders with PCA or Principle Component Analysis to identify digits that are handwritten. It concluded that accuracy is increased when autoencoders are used while PCA provides performance benefits. This research prioritizes accuracy over performance, therefore autoencoders were preferred for dimension reduction for images of Hindi characters as well as the images for mnemonics. This research implemented convolutional autoencoders for the obtaining the results, however, other types of autoencoders were also considered. The convolutional autoencoder came out with the best of results.

3 Research Methodology

This research has been implemented along the traces of CRISP-DM with less inclination towards the business side. CRISP-DM has 6 stages in total, including Business understanding, Data understanding, Data preparation, Modelling, Evaluation and Deployment. For the research, it was decided to drop the first and last stage because it is less business and product oriented as of now. The research was a skeleton for a possible product like an e-learning application where the business concerns and deployment come into question. This research was more about promoting the idea of creation of visual mnemonic via machine learning techniques rather than creating them manually. Hence, the core concept was focused upon in the research utilizing the middle four stages of CRISP-DM.

CRISP-DM was an appropriate choice because it includes spiralling to and fro between the Data preparation and Modelling phases which was an important aspect of this research. The data had to be manipulated based on the outputs received by the models trained to attain the most suitable dataset. Some modifications like dataset size, image type, image class, tuning of various parameters had to be done repeatedly to obtain the favourable output.

Figure 1 provides a brief idea of the steps followed by the research. The succeeding sections explains these steps in details.

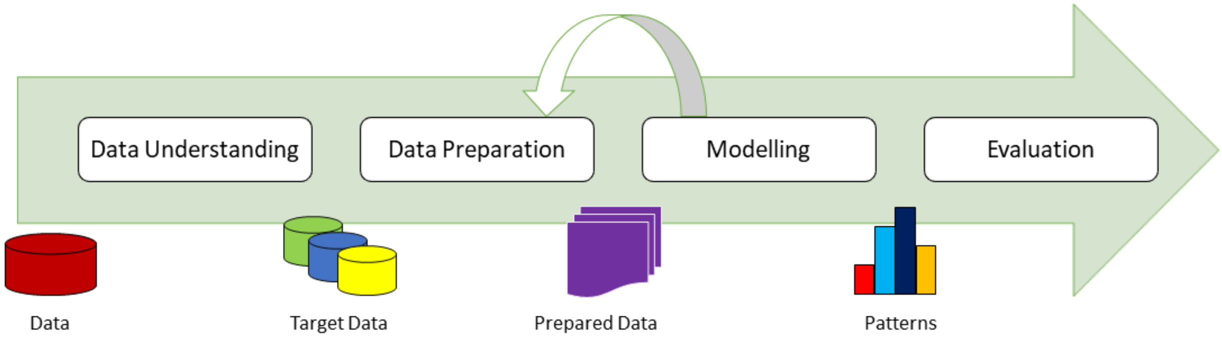

Figure 1: Research Methodology

3.1 Data Understanding

There were various phases involved in the understanding of the data. Once the purpose of the research was clearly understood, the data that was appropriate and required for the research was collected. As explored, the research required more than one dataset to be considered in order to fulfil its aim. The datasets used are explained further in the following subsections.

3.1.1 Devanagari Font Characters

Since the research is about the Devanagari characters (Hindi script is known Devanagari), the research had a small dataset (as shown in *Figure 2*) of the computerized Hindi script font characters. This dataset has a total of 46 images comprising of 36 Hindi alphabets and 10 numeral digits.

Figure 2: Devanagari/Hindi Font Characters

The dataset was constructed from one of the most widely used font for Hindi, namely Akshar. The font characters were converted into images in png format using a Python code.

3.1.2 Handwritten Devanagari Characters

Another dataset used by research is a dataset containing handwritten Devanagari characters(Jha, 2018). A collection of handwritten Devanagari digits and alphabets (as shown in *Figure 3*) is included in the dataset. Approximately 9.6k pictures of 10 numeral digits and 36 Hindi alphabets are contained in the dataset. Figure 3 shows a preview of the different handwritings available in the dataset.

Figure 3: Devanagari/Hindi handwritten Characters

It is a publicly available dataset comprising of grayscale images, which could easily be found on Kaggle. Each character in the dataset has a resolution of 32x32, 28x28 being the size of character and 2-pixel padding on the sides.

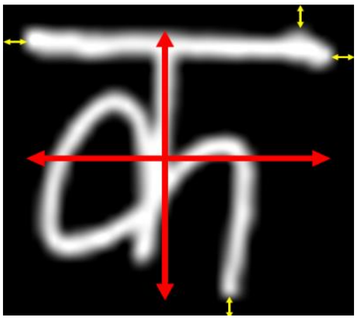

Figure 4: Character padding for the Dataset 2

Figure 4 illustrates the character resolution. Red arrows indicate the character size of 28x28 whereas yellow arrows show the 2x2 padding on all sides. This would support the model to train itself better in order to identify the characters effectively.

3.1.3 Mnemonic image datasets

One of the most crucial aspect of this research was to get a varied and appropriate set of images that would prove to be impactful as Mnemonics. The image dataset had to be updated throughout the research to make it more efficient. It was formed by integrating several datasets of different things. The resulting dataset has images ranging from Bikes and Taps to various animals and even Pokemons. Some of the images were collected manually, resized and formatted accordingly to create a dataset that would be valuable for the research.

The initial dataset had nature images that was obtained from Kaggle (Roy, Bhattacharya and Ghosh, 2018) . Upon various cycles of training and analysing results¸ it was found that some of the images in the dataset were not contributing as much to the results, so they were removed and replaced. Among 8 classes of this dataset, 3 classes were utilized for this research. *Figure 5* shows the entire dataset that was finally used in the research.

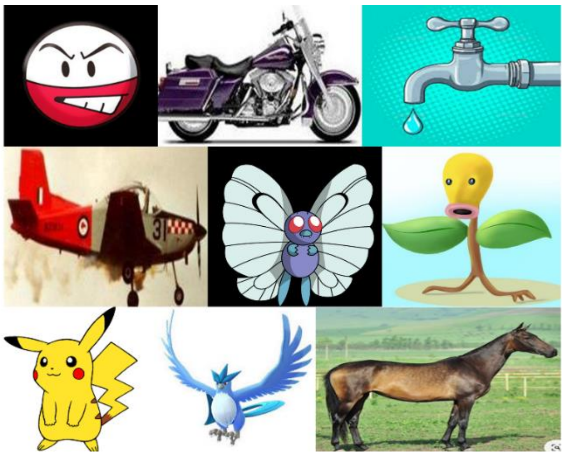

Figure 5: Potential Mnemonic Images

Another dataset used was a Pokemons dataset (Lance Zhang, 2019) . The dataset has imaginary creatures from a popular anime. It was chosen to spike interest of the youth who could benefit from Mnemonical learning. From this dataset as well, some of the images were too complicated and didn't serve much purpose to the model. A portion of the mnemonic dataset was also gathered from Google images, keeping in mind the privacy and confidentiality issues. In general, 25 images of each class were utilized to train the model.

3.2 Data Preparation

The preparation of data was a trivial part of this research since the decision of what and how to prepare was considerably complicated. After the selection and basic exploration of data, the data needed to be prepared for applying to the models and implementing the research. The Devanagari font characters and the Devanagari handwritten characters didn't require a huge amount of preparation. The latter was greyscale and in uniform size to be used for the research. Similar was the case with the font characters.

However, the mnemonic dataset had to undergo heavy preparation. Since the dataset was gathered and integrated from numerous sources, a uniformity had to be maintained among the different image sets. The following tasks were performed to prepare the data for training autoencoder: -

1. **Data resize** - All the images were resized into a size of 32x32 pixels to be used for training CNN. For autoencoders, the images were resized to a size of 64x64 pixels.

2. **Data type convert** - The images gathered from various sources was in different image formats. Therefore, research converted all the other image types to PNG images. This was done with the help of a small Python program to ease the process. The images were loaded in RGBA mode for training.

3. **Dataset reduction** - Autoencoders reconstructs the input image that is given to it in reduced dimension. This requires a lot of processing, which is why the process can be extremely slow and complicated. Hence, an appropriate number of images for each data label was chosen. Each class has about 25 to 30 images which initially was up to 100 images per class for Mnemonic dataset and 300 images per class for handwritten Hindi dataset.

It should, however, be noted that entire process of data preparation was continuous throughout research. The Mnemonic dataset had to updated upon analysing the output.

3.3 Modelling and Results

As previously mentioned, there was a lack of considerable existing literature for this research. Therefore, an approach was fabricated to carry out the research in 4 stages of implementation. Each of the stages had their own objectives, preparation, implementation, processing, results and evaluation. Every stage had its own proposed architecture. The research explored the existing state of art to attain a better perspective of the technique to be used for individual stages. As explored and observed in the previous works, CNN was chosen to carry out the initial phase of the research. The task of recognition of handwritten characters was appropriate for CNN since it has been used extensively for this task. CNN, however, efficient couldn't have achieved the results for this research unaccompanied. The research required the assistance of an algorithm that could ease the process of mapping potential mnemonic images to the character set. Hence, Autoencoders were utilised for feature extraction of both character set as well as the images dataset.

The research had to go back and forth from this stage to the earlier one, i.e. Data preparation. It was a dynamic process for this research since the choice of the Mnemonic image to be chosen couldn't be made at once without looking up the output. There had to be various changes and enhancements done in the image dataset to attain suitable outputs.

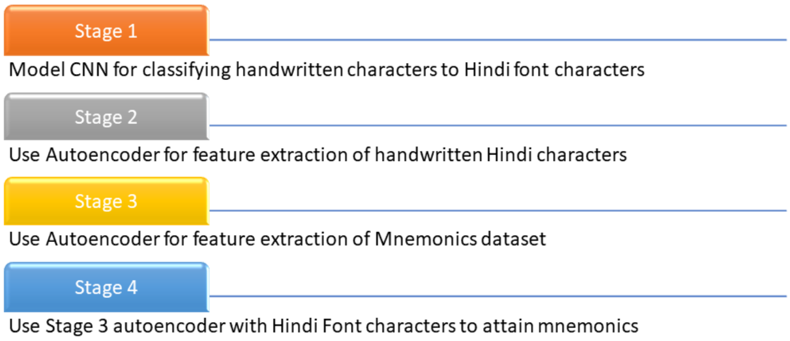

Figure 6: Approach for research implementation

The research used Tensorflow and Keras to implement deep learning. A lot of challenges were faced while using the aforementioned libraries. There are several incompatibilities among the versions which makes it time consuming to debug and code accordingly. For this research, the versions for these libraries had to downgraded to get everything in sync. The details are included in the configuration manual.

The 4 stages, as shown in *Figure 6*, on which the research is based on are explained in the following subsections. Also, the architecture, implementation and result of each of the stage would be elaborated in detail.

3.3.1 Stage 1: CNN classification of handwritten Devanagari characters

This research aims to promote language learning using Mnemonics. Hence, the initial stage was to identify a character that could be given as an input by the user. The model trained in this stage identifies the input Hindi handwritten character and classifies it as one of the Hindi font characters. The stage uses CNN for identifying and classifying handwritten Hindi characters. CNN has a proven credibility in terms of identifying and classifying handwritten language characters. A large proportion of existing literature (Acharya, Pant and Gyawali, 2016) (Arif *et al.*, 2018) (Neri *et al.*,

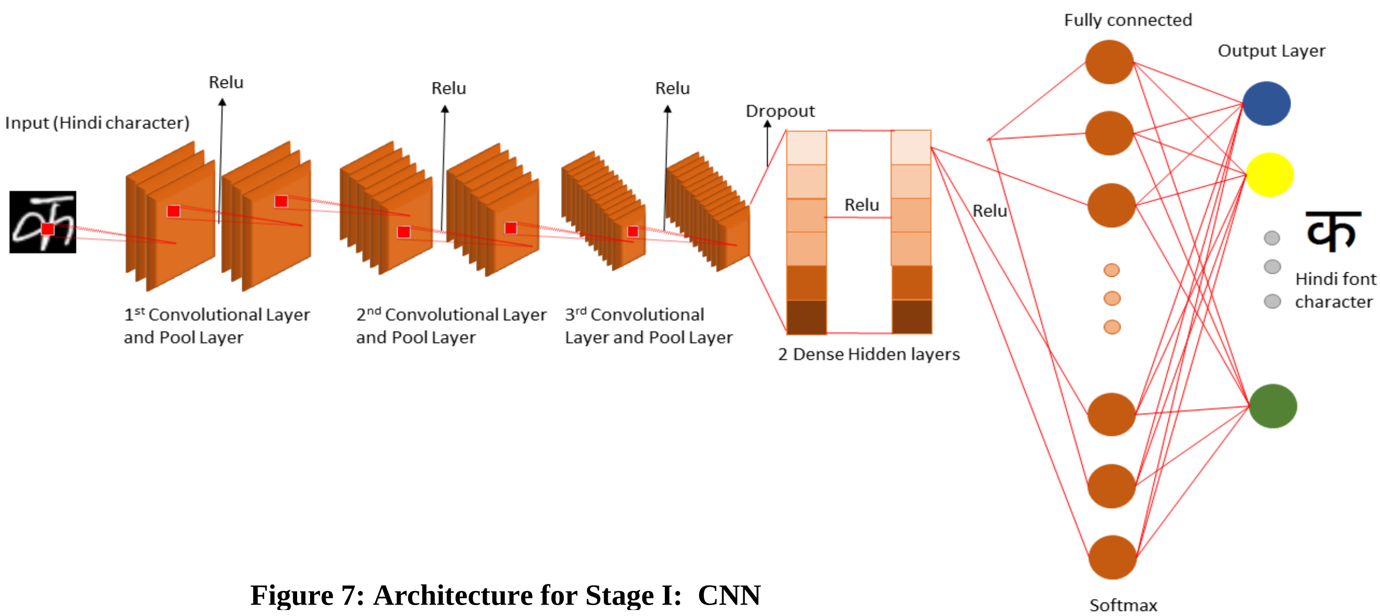

Figure 7: Architecture for Stage I: CNN

2020) (Wen, Shao and Zheng, 2019) (Shopon, Mohammed and Abedin, 2016) on handwriting recognition were implementing a CNN model.

The research utilized an architecture containing 3 convolutional layers. Each of the convolutional layer is followed by a pooling layer with max pooling. Further, 2 hidden layers are added and are densely connected with 128 neurons. The output layer is also fully connected with 46 neurons in it. The neurons used were 46 to represent each of the characters of the Hindi language. The activation function used between the convolutional and pooling layers and between the hidden layers is ReLU. Softmax is used in the last layer. *Figure 7* illustrates the architecture of the CNN used. A dropout of 0.25 was added before the first hidden layer. The CNN model was trained for 40 epochs. At the end of each epoch, accuracy of the model was observed. The final accuracy came out to be 98.48%, which is pretty decent. This stage, however, was just the initial step of the research and is to set up the base ground of the actual purpose of the research.

3.3.2 Stage 2: Autoencoder I to reconstruct Devanagari characters

Autoencoders play an important part in this research. Autoencoders have been utilized to extract essential features from the datasets considered, namely the handwritten Devanagari character set and the Mnemonic images dataset. This enabled the important features to be captured by the Autoencoders. The autoencoder reconstructed the given input with the assistance of the features extracted. In order to select the type of autoencoder to be used, the existing literature was studied. Upon analysing¸ it was observed that simple autoencoders and the convolutional autoencoders were among on the higher scale of the popularity status. Initially, simple autoencoder was trained for this research. However, due to dissatisfaction of results, the simple autoencoder was discarded for the convolutional autoencoder.

The convolutional autoencoder architecture consists of an encoder and a decoder. The total number of layers used in this stage autoencoder were 8, 4 layers for encoder and 4 layers for the decoder. Encoder consisted of 2 convolutional layers with a dropout of 0.2 after them. The image was then flattened into vector. It was further followed by 2 hidden layers with a dropout of 0.2 between them. The decoder had 4 layers of which 2 layers are hidden layers and 2 layers are transposed convolutional layers. Starting from hidden layers, the input was reshaped and fed to transposed convolutional layers. *Figure 8* illustrates the architecture of the CNN used. The activation function used is elu. 128 neurons are used for the main layers of encoder and decoder. The number of epochs was 20. These values were used in Stage 3 autoencoder as well.

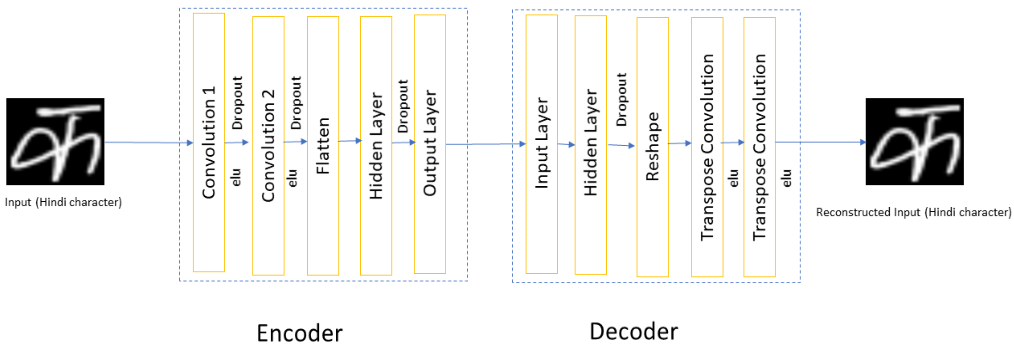

Figure 8: Architecture for Convolutional autoencoder for Digit/Character recognition

Before training the autoencoder, dataset was created and explored. *Figure 10* shows the random images from dataset used by the autoencoder. The convolutional autoencoder was trained using png images as input. It then reconstructed the given input images by learning its substantial features. This resulted in elimination of less significant features. *Figure 9* depicts the output obtained from Autoencoder I.

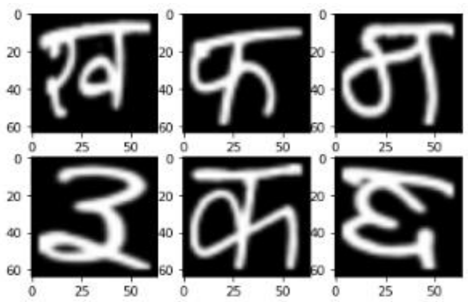

Figure 10: Data exploration for Stage

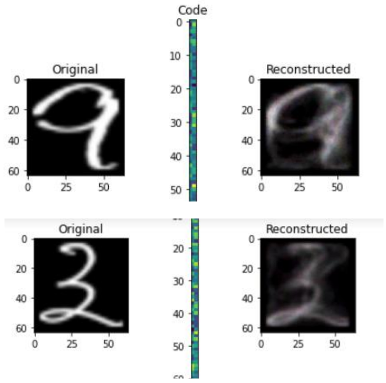

Figure 9: Output for Stage 2

The autoencoder in Stage 2 consumed the Devanagari handwritten characters for pulling out the beneficial features and reconstruct the characters. This stage of the research holds importance because the succeeding stages are directly or indirectly dependent on this. This stage facilitated the decision of the tuning of parameters that would result in the most suitable output. The implementation of this stage was required once to attain the values of the tuning parameters like number of layers, number of neurons, activation function, dropouts among others.

3.3.3 Stage 3: Autoencoder II to reconstruct image dataset

This stage utilized a convolutional autoencoder as well. However, the autoencoder in this stage was trained on the image dataset that had been prepared to be the Mnemonic images. *Figure 12* shows the random images from dataset used by the autoencoder. The importance of Stage 2 was reflected in Stage 3 when the same set up of the parameters was used to train the Autoencoder II. Autoencoder I gave the best possible setting for extracting features from the Devanagari handwritten character set, hence Autoencoder II was established with the same setting to train image dataset. *Figure 11* depicts the output obtained from Autoencoder II in Stage 3.

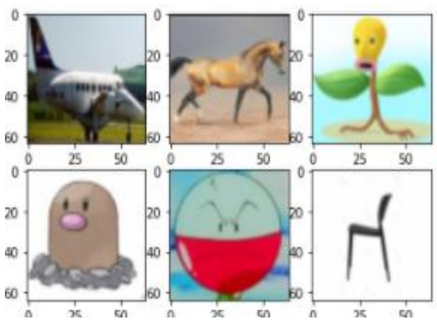

Figure 12: Data exploration for Stage 3

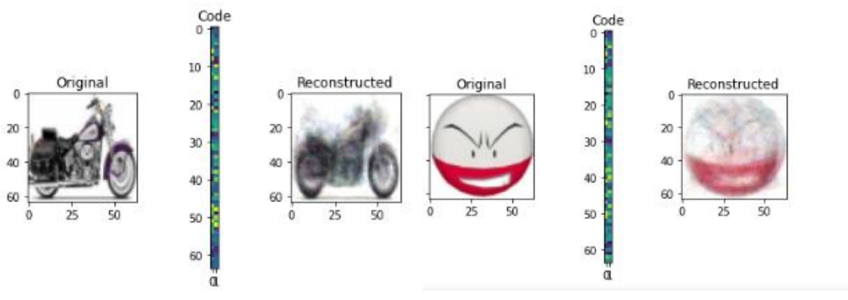

Figure 11: Output for Stage 3

3.3.4 Stage 4: Autoencoder II with Hindi Font characters to attain Mnemonics

The final stage of the research was to make use of the Stage 3 autoencoder. The dataset of the Devanagari font characters was provided as input to Autoencoder II which was already trained with the image dataset created to become Mnemonics. The autoencoder gave a reconstructed image as an output for each of the Hindi font characters that was fed to it.

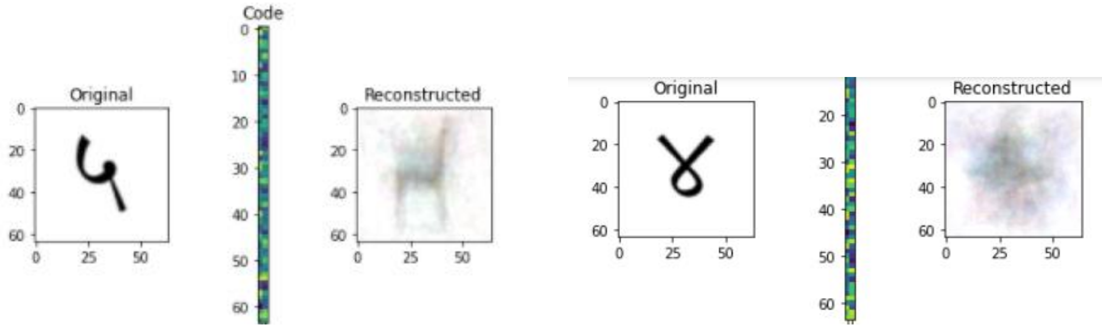

Figure 13: Output for Stage 4

Figure 13 shows the outputs obtained in Stage 4. Some traces of the strokes could be seen on the right side of the code but it is hard to make out what the reconstructed image is for the character. Hence, these results are better analysed in the Evaluation phase of the research.

3.4 Evaluation

Since the research had multiple phases and stages, the evaluation of each stage had to be done individually. There were two machine learning models used in the research, namely CNN and Autoencoder. Therefore, different methods were identified to analyse and evaluate the results of both the algorithms. This is explained in detail in the following sections.

3.4.1 Evaluation of CNN Model (Stage 1)

The CNN Model used to identify and classify handwritten Hindi characters has been evaluated using various methods. Each epoch gave the evaluation results for each of the methods. The evaluation methods used were: -

a. Accuracy – The final accuracy for Stage 1, i.e., the CNN part of the research came out to 98.48%. This was the accuracy at the end of 40 epochs. *Figure 14* shows a graph of the variation of accuracy with each epoch.

b. Loss – The final loss of the model was 0.0529, which is significantly low.

c. Value Loss – The final value loss of the model was 0.0809, which is a low value too.

d. Value Accuracy - The final value accuracy of the model was 0.9798.

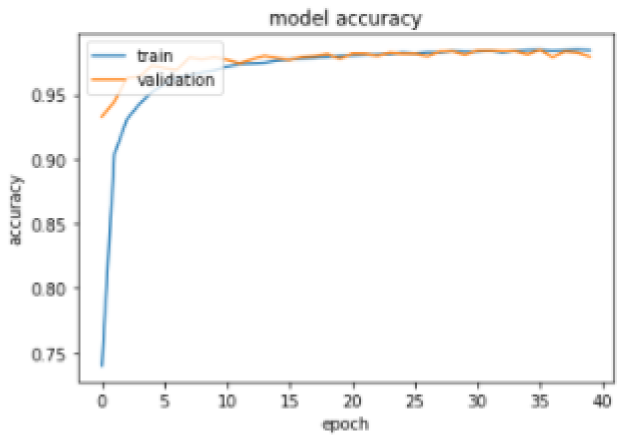

Figure 14 : Accuracy vs Epoch graph for CNN

3.4.2 Evaluation for Autoencoders (Stage 2 and 3)

Autoencoders, precisely Convolutional autoencoders, have been utilized by 2 stages (Stage 2 and Stage 3) of the research. Since, Stage 2 was to prepare for Stage 3, only the results for Stage 3 have been recorded for evaluation. The methods used to evaluate were: -

a. Loss: The final loss of the Autoencoder II was 0.0445, which is considerably low.

b. Value Loss: The final value loss of the Autoencoder II was 0.0384, which is a value on the lower side of the scale.
c. MSE Score: The MSE score for Autoencoder II was 0.038. The lower the MSE score, the better.

3.4.3 Evaluation for Mnemonics (Stage 4)

As mentioned earlier, the potential mnemonic images were mapped to the Hindi characters in Stage 4. It is challenging to decide and measure the suitability of a particular Mnemonic for its respective character. A certain amount of human involvement is required to choose the image that could easily imprint on a learner's mind and help in memorizing the character. The research, however, evaluates the reconstructed image presented as the output in Stage 4. The nearest neighbours for that distance are identified by measuring the Euclidean distance between the predicted image and the potential Mnemonic dataset. It is done using the Nearest Neighbours algorithm. The algorithm evaluated the distance between the output image and each of the images from the dataset. It then organized the images in increasing order of their distances. These images work as the suggested potential mnemonics for the Hindi characters.

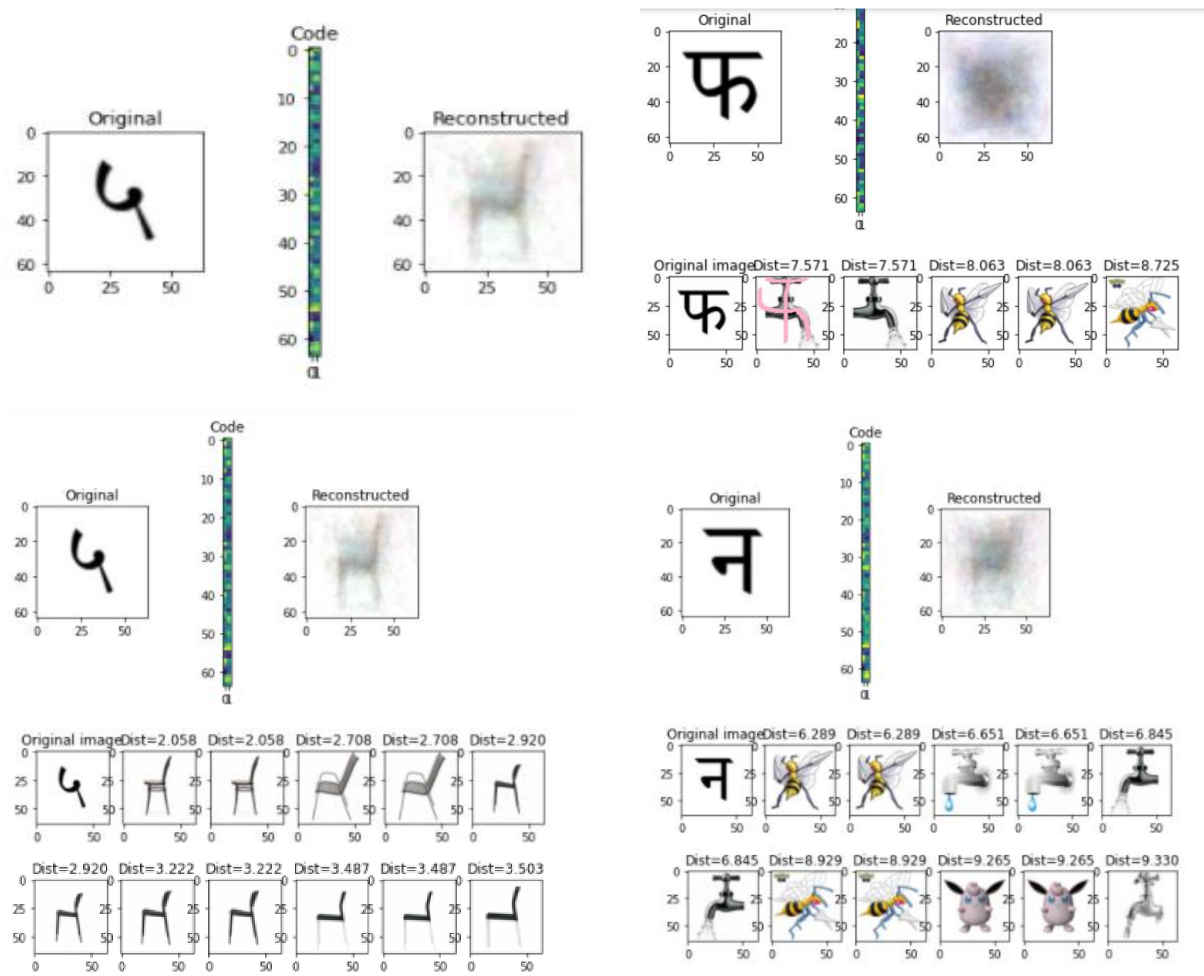

Figure 15: Evaluated results for Mnemonic images

Figure 15 displays the evaluation of the mnemonic images. The numbers shown above each suggested image is the Euclidean distance between the reconstructed image and that particular image. However, it isn't essential that only the closest image can be a better mnemonic. It depends on the individual's perception.

4 Discussion

As per the findings, a research like this has been implemented for the first time. In the past, all the researches that are done on visual mnemonics and language learning involve manual creation of mnemonics. The results obtained are observed and analysed for the first time. Nevertheless, the results provided by the research were satisfactory but have a great potential to improve as well. CNN provided pleasant results with a competitive accuracy. Dataset 2, i.e. handwritten Devanagari Character set was initially used by (Acharya, Pant and Gyawali, 2016) for Hindi character recognition using CNN. Their final accuracy turned out to be 98.47% while this research achieved 98.48%, thereby upping the benchmark by 0.01%. CNN was analysed by random test images to predict the output and display the Hindi font character as prediction. 9 out of 10 times it provided the correct output. Therefore, CNN phase of the research has an agreeable validation based on the benchmark taken for this research.

Autoencoder I reconstructed the handwritten characters quite well. The output was up to the expectations and one could recognize the input character by looking at the reconstructed output. Autoencoder II recreated the input image acceptably as well. However, some of the images in the dataset had colourful backgrounds and certain noise in it, so the results reflected were more distorted than the others. The dataset fed to Autoencoder I had black background, as shown in *Figure 17*; hence, the distortion and blurriness wasn't evident there. Even with the complicated structures and colours as seen in *Figure 16,* Autoencoder II reconstructed the images satisfactorily.

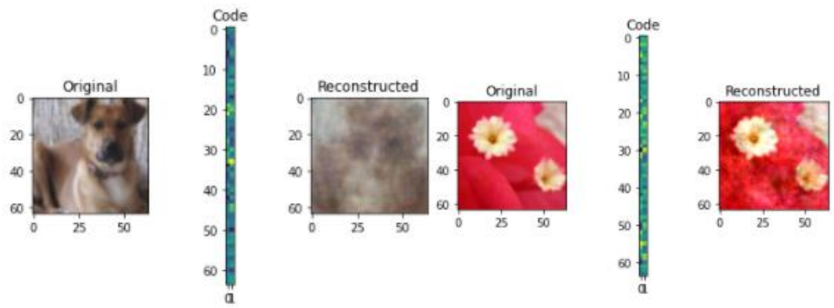

Figure 16: Complicated images for Autoencoder II

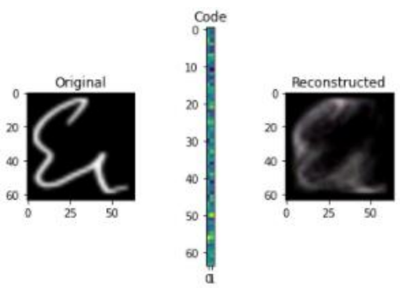

Figure 17: Black background images for Autoencoder I

Hindi script, Devanagari has 46 characters in total, 36 alphabets and 10 numerals. As it can be observed in the results, same images might be suggested for more than 1 character. 46 characters equals to 46 groups of images, presuming one for each character. The gathering and training of 46 classes of images is a challenging task. Therefore, the generated mnemonics for some of the characters in this research are repeating. The potential mnemonic image set is almost half the number of total characters, which is a shortcoming of the research. This was an important grasp from the research and would be kept in mind in future.

The suggested images that came out in the evaluation of Stage 4 were extremely helpful in decrypting the results. By looking at the final suggestions and the character, one could identify on how the autoencoder worked and why the particular images were chosen. Some of the strokes and structure of the images matches those of the characters. However, a lot of Hindi characters and numerals appear similar to each other. This could be another reason why there wasn't strong distinguishing among them and the images repeated themselves.

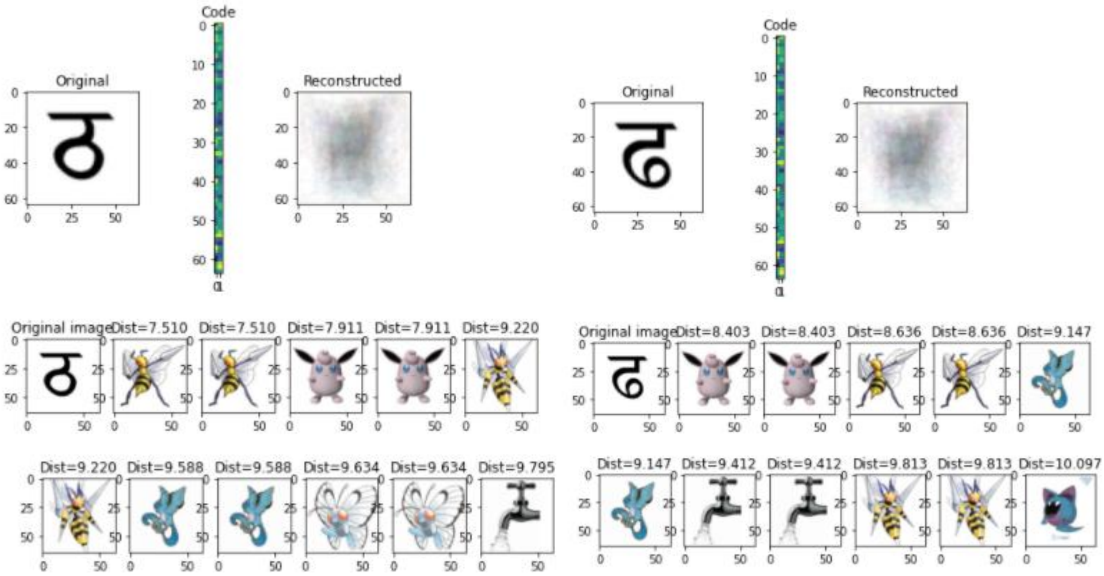

Figure 18: Example of Hindi characters with similar structure

For instance, the characters in *Figure 18* depict the results obtained for characters with similar structure and shape. The characters on right and left have similar structure, i.e. a tiny vertical stem attached to a rounded and curled structure. However, the character in the left has a closed circle (th-) whereas the one on the right (ḍh-) has open circle which has another enclosing round structure. The results implied that the autoencoder used to generate the output understood the high-level structure of the character, i.e. the vertical stem and the bottom circle. The reconstructed output is matched with the entire dataset and the resulting images are analysed and evaluated.

It could be observed that the results include similar images for both the characters. The first result of the character in left, the bee-like image (with Dist = 7.510), has a tiny stem (Bee's antenna) and a rounded body. The third result, the rabbit-like image (with Dist = 7.911), has ears as the vertical stem and a big rounded body. These, along with the other results in this, are common for both the characters. Therefore, it could be analysed and understood on how the autoencoder worked and repeated the results for similar looking characters.

5 Conclusion and Future work

The purpose of the study was to reduce the manual labour in the process of creating Mnemonics for the characters of a language. Hindi doesn't have a standing research for visual mnemonics; hence this research was carried out using Hindi. Visual mnemonics for the characters of Hindi language were successfully generated using machine learning algorithms implemented in 4 stages. The results obtained were overall satisfactory. The techniques used were CNN and Convolutional Autoencoders. CNN gave an accuracy of 98.48% whereas Autoencoder gave a final MSE score of 0.038. Mnemonic images reconstructed by the autoencoder weren't identifiable by naked eyes, therefore, the images were evaluated using nearest neighbour algorithm. A set of images was suggested as an output that could act as visual mnemonics for Hindi characters. The efficiency of the generated mnemonic is dependent on the individual user. If the output image impacts one's mind, they could memorize it to recall the structure and shape of the character.

Since this was a first time attempt at generating visual mnemonics using machine learning, there is a huge scope of enhancements and improvements. Firstly, the research could be carried out with a large variety of classes for potential mnemonic images, probably twice the number of characters. Secondly, the research could be carried out using Generative Adversarial Networks (GAN) instead of autoencoders for reconstruction of images. Thirdly, an extension for this research could be the generation of audio mnemonics for Hindi as well.

Acknowledgement

I wish to extend my heartiest gratitude towards the people who made this research possible. I would like to thank Dr. Vladimir Milosavljevic for supporting me to attempt a fairly new research and guiding me through the tough times. A huge thanks to Dr. Michael Bradford for providing directions and guidance for the research implementation. I am ever thankful to my family and friends for their constant motivation and encouragement.

References

7,000 Labeled Pokemon | Kaggle (2019). Available at: https://www.kaggle.com/lantian773030/pokemonclassification (Accessed: 16 August 2020).

Acharya, S., Pant, A. K. and Gyawali, P. K. (2016) 'Deep learning based large scale handwritten Devanagari character recognition', in *SKIMA 2015 - 9th International Conference on Software, Knowledge, Information Management and Applications*. Institute of Electrical and Electronics Engineers Inc. doi: 10.1109/SKIMA.2015.7400041.

Alani, A. A. (2017) 'Arabic Handwritten Digit Recognition Based on Restricted Boltzmann Machine and Convolutional Neural Networks', *Directory of Open Access Journals*, 8(4), p. 142.

Almotiri, J., Elleithy, K. and Elleithy, A. (2017) 'Comparison of autoencoder and Principal Component Analysis followed by neural network for e-learning using handwritten recognition', in *2017 IEEE Long Island Systems, Applications and Technology Conference (LISAT*. Farmingdale, NY, USA.

Arif, R. B. *et al.* (2018) 'Study and Observation of the Variations of Accuracies for Handwritten Digits Recognition with Various Hidden Layers and Epochs using Convolutional Neural Network', in *2018 4th International Conference on Electrical Engineering and Information & Communication Technology (iCEEiCT*. Dhaka, Bangladesh.

Bahrami, Z. N., Izadpanah, S. and Bijani, H. (2019) 'The Impact of Musical Mnemonic on Vocabulary Recalling of Iranian', *International Journal of Instruction*, 12(1), pp. 977–994,.

D'Angelo, G., Ficco, M. and Palmieri, F. (2020) 'Malware detection in mobile environments based on Autoencoders and API-images', *Journal of Parallel and Distributed Computing*, 137, pp. 26–33.

Fu, H. *et al.* (2019) 'Improved semi-supervised autoencoder for deception detection', *PLoS ONE*, 14(10), pp. 1–13.

Gunasekaran, A., McNeil, R. D. and Shaul, D. (2002) 'E-learning: Research and applications', *Industrial and Commercial Training*, 34(2), pp. 44–53.

Haydon, T., Musti-Rao, S. and Alter, P. (2017) ‘Comparing Choral Responding and a Choral Responding Plus Mnemonic Device During Geography Lessons for Students with Mild to Moderate Disabilities’, *Education and treatment of children*, 40(1), pp. 77–95.

Jain, P., Darbari, H. and Bhavsar, V. C. (2017) ‘Cognitive support by language visualization: A case study with hindi language’, in *2017 2nd International Conference for Convergence in Technology (I2CT*. Mumbai, India.

Jha, S. (2018) ‘Kaggle’. Available at: https://www.kaggle.com/jhashanku007/devnagri-hindi-dataset.

Kalbhor, S. R. and Deshpande, A. M. (2018) ‘Digit Recognition Using Machine Learning And Convolutional Neural Network’, in *2018 2nd International Conference on Trends in Electronics and Informatics (ICOEI*. Tirunelveli, India.

Kimmel, L. H. (2020) ‘Hindi Becomes India’s Dominant Language’. Available at: http://search.ebscohost.com/login.aspx?.

Kumar, A. S. and Nair, J. J. (2019) ‘Pair wise training for stacked convolutional autoencoders using small scale images’, *Journal of Intelligent & Fuzzy Systems*, 36(3), pp. 1987–1995,.

Leelanupab, T. and Anonthanasap, O. (2017) ‘Keyword mnemonic and retention of second-language vocabulary words’, in *18th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD*. Kanazawa, Japan.

Lin, E., Mukherjee, S. and Kannan, S. (2020) ‘A deep adversarial variational autoencoder model for dimensionality reduction in single-cell RNA sequencing analysis’, *BMC Bioinformatics*, 21(1), pp. 1–11.

Lundin, L. L. (2019) *E-Learning*. Salem Press Encyclopedia.

Mahaffey, A. L. (2020) ‘A Learning Tool for Chemistry and Health Professions Students: Mnemonics for Writing Net Ionic Equations’, *Journal of College Science Teaching*, 49(3), pp. 27–30.

Narang, S., Jindal, M. K. and Kumar, M. (2019) ‘Devanagari ancient documents recognition using statistical feature extraction techniques’, *Springer*, 44(6), pp. 1–8.

Neri, M. C. G. *et al.* (2020) ‘A Convolutional Neural Network for Handwritten Digit Recognition’, *International Journal of Combinatorial Optimization Problems and*, 11(1), pp. 97–105.

Reddy, R. V. K., Rao, B. S. and Raju, K. P. (2018) ‘Handwritten Hindi Digits Recognition Using Convolutional Neural Network with RMSprop Optimization’, in *2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS*. Madurai, India.

Rohland, L. (2019) *Mnemonic*. Salem Press Encyclopedia of Health.

Roy, P., Bhattacharya, S. and Ghosh, S. (2018) ‘Kaggle’. Available at: https://www.kaggle.com/prasunroy/natural-images.

Shopon, M., Mohammed, N. and Abedin, M. A. (2016) ‘Bangla handwritten digit recognition using autoencoder and deep convolutional neural network’, in *2016 International Workshop on Computational Intelligence (IWCI*. Dhaka, Bangladesh.

Siddique, F., Sakib, S. and Siddique, M. A. B. (2019) ‘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.

Song, J. *et al.* (2019) ‘Alphapwd: A Password Generation Strategy Based on Mnemonic Shape’, *IEEE Access*.

Sufian, A. *et al.* (2020) ‘BDNet: Bengali Handwritten Numeral Digit Recognition based on Densely connected Convolutional Neural Networks’, *Journal of King Saud University - Computer and Information Sciences*.

Tamara, A., Rusli, A. and Hansun, S. (2019) ‘Utilizing E-learning with Pattern Recognition and Mnemonic to Enhance Japanese Characters Learning Experience’. Tokyo, Japan.

Wang, Z. *et al.* (no date) ‘Fault diagnosis of air-conditioning refrigeration system based on sparse autoencoder’, *International Journal of Low Carbon Technologies*, 14(4), pp. 487–492.

Wen, Y., Shao, Y. and Zheng, D. (2019) ‘A Novel Deep Convolutional Neural Network Structure for Off-line Handwritten Digit Recognition’, in *ACM International Conference Proceeding Series*.

Yadav, M. and Bhojane, V. (2019) ‘Semi-Supervised Mix-Hindi Sentiment Analysis using Neural Network’, in *2019 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence) Cloud Computing*. Noida, India.

Ying, Y., Rawendy, D. and Arifin, Y. (2016) ‘Game Education for Learning Chinese Language’. Jakarta, Indonesia.

Zhan, H. *et al.* (2019) ‘CNN-based Hindi Numeral String Recognition for Indian Postal Automation’, in *2019 International Conference on Document Analysis and Recognition Workshops (ICDARW*. Sydney, Australia.