Neuroscience Informatics 2 (2022) 100016

Contents lists available at ScienceDirect

Neuroscience Informatics

www.elsevier.com/locate/neuri

NaaS

# Digitization of handwritten Devanagari text using CNN transfer learning – A better customer service support

Sandeep Dwarkanath Pande [a,b,*], Pramod Pandurang Jadhav [c], Rahul Joshi [d], Amol Dattatray Sawant [e], Vaibhav Muddebihalkar [f], Suresh Rathod [d], Madhuri Navnath Gurav [b], Soumitra Das [g]

[a] Dr. D Y Patil Institute of Technology, Pimpri, Pune, MH, India
[b] Agasti Publishers, Talegaon Dabhade, Pune, MH, India
[c] Department of Computer Engineering, GH Raisoni Institute of Engineering and Technology, Wagholi, Pune, MH, India
[d] Symbiosis Institute of Technology, Symbiosis International (Deemed University), Pune, MH, India
[e] Vishwakarma Institute of Technology, Pune, MH, India
[f] Doshaheen Solutions Pvt Ltd, Pune, MH, India
[g] Department of Computer Engineering, Indira College of Engineering and Management, Pune, MH, India

ARTICLE INFO

*Article history:*
Received 21 September 2021
Received in revised form 1 October 2021
Accepted 1 October 2021

*Keywords:*
Devanagari image to text
Transfer learning
Devanagari text digitization
CNN for Devanagari text recognition
Alexnet

ABSTRACT

Devanagari script is one of the bases of various language scripts in India. With the growth of computing and technology, manual systems are replaced by automated one. The purpose of this research is to automate the existing manual system for digitization of Devanagari script with the use of an automated approach so that it saves time, antique data. The prescriptions given by the expert doctors and the treatments which are present in ancient Vedic literature are useful for handling patients with serious diseases. Digitization helps in easy access, manipulation, and longer storage of this data. Unlike Western languages such as English, Devanagari, is a famous script in India which does not have formal digitization tools. This work employs the best suited techniques that are useful to enhance the recognition rate and configures a Convolutional Neural Network (CNN) for effective Devanagari handwritten text recognition (DHTR). This approach uses Devanagari handwritten character dataset (DHCD) which is a vigorous open dataset with 46 classes of Devanagari characters and each of this class has two thousand different images. After recognition, conflict resolution is subtle for effective recognition therefore, this approach provides an arrangement to the user to handle the conflicts. This approach obtains promising results in terms of accuracy and training time.



## 1. Introduction

Various countries in the world are adopting the process of digitization and promoting the structure and infrastructure of various fields to support digitization. It is said that the digitization is the pathway to prosperity. Digitization increases greater transparency and efficiency. In a country like India, several organizations such as government offices, academic institutes, agricultural institutes, National Informatics Centre (NIC), courtrooms, registrar offices and small-scale vendors maintains various documents which are available in either handwritten or printed form in Devanagari script. Furthermore, the ancient Vedic literature, medical prescriptions to the patients and the diagnosis summary are also present in Devanagari script. Huge number of such documents are required to be conserved. Most of the languages in India including Hindi are based on Devanagari script. Therefore, there is a huge demand for a technology that transforms Devanagari handwritten text into a digital text. Offices such as NIC deals with issues like storage of enormous volumes of data. This data is conventionally stored in the forms documents that take a lot of physical space which is expensive and limited. Searching and indexing is required to find data, which itself is a tedious and cumbersome task. The quality of the papers and ink used to write or print the data gets deteriorated after some years. This makes it very difficult to read and interpret the information present on these documents.

* Corresponding author at: F. No.: 203, White House, Bhairavnath Nagar, Pimple Gurav, Pune 411061, Maharashtra, India.

*E-mail addresses:* sandeep7887pande@gmail.com (S.D. Pande), ppjadhav21@gmail.com (P.P. Jadhav), rahulj@sitpune.edu (R. Joshi), amoldsawant@gmail.com (A.D. Sawant), vaibhav.muddebihalkar@gmail.com (V. Muddebihalkar), sureshrathod1@gmail.com (S. Rathod), guravmadhuri93@gmail.com (M.N. Gurav), reach.soumitra.das@gmail.com (S. Das).

https://doi.org/10.1016/j.neuri.2021.100016

**Table 1**
Basic consonants in Devanagari.

| | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| क | ख | ग | घ | ङ | च | छ | ज | झ | ञ | ट | ठ | ड | ढ | ण |
| त | थ | द | ध | न | प | फ | ब | भ | म | य | र | ल | व | श |
| ष | स | ह | ळ | क्ष | ज्ञ | | | | | | | | | |

**Table 2**
Vowels in Devanagari.

| | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| अ | आ | इ | ई | उ | ऊ | ए | ऐ | ओ | औ | अं | अः | अॅ | ऑ | ऋ |

**Table 3**
Digits in Devanagari.

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| ० | १ | २ | ३ | ४ | ५ | ६ | ७ | ८ | ९ |

**Table 4**
Consonant clusters formed by combining consonants and vowels in Devanagari.

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| ध | धा | धि | धी | धु | धू | धे | धै | धो | धौ | धं | धः |

**Table 5**
Special consonant clusters formed by combining raphars with consonants and two consonants combination forming a consonant cluster in Devanagari.

| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| कृ | क्र | र्क | ट्र | ऱ्ह | त्र | श्र | ऱ्क | रँ | त् | ॐ | स्क | ग्ध | क्य | च्छ | रु | रू |

Ancient documents have precious information, which needs to be preserved and digitized. As ancient documents have mentioned the process for treating and curing mentally retarded patients, which includes meditation, yoga, and other Ayurveda practices. Most of this information is in Devanagari form [1]. It is a tedious task to collect all this information and generate a proper course of treatment for effectively handling the mentally ill patients. This information can also be further used for constructive research. Most of the population in India stays in rural place, and the prescriptions and reports generated by the doctors are in Devanagari form. This information is needed to be transformed so that, it can be incorporated into further analysis. Since the storage of documents is in a physical place, various accidents like fire, flood, tsunami, earthquake or other natural misfortunes can happen. Manual manipulation of the data in physical form and proof tinkering is possible in such cases.

There are various technologies designed as a standard for digitizing a printed text in an ideal format, but there is a lack of applications to digitize the handwritten document with acceptable accuracy. If the digitization of DHTR is successful, it will help to save the data for a longer period. Typically, thousands of files that require a huge space approximately a room for storage, will take up about 1 TB of digital capacity. It would be suitable to transfer data from one place to another; the data can be randomly accessed/extracted from any location with proper authorization. Digitization of data helps to reduce the possibility of evidence tampering as the digital medium keeps a full-proof record of the transactions being made. Essential documents can be distributed and saved at various places so that it becomes challenging for an unauthorized user to access and modify the data. This is why an effective and efficient DHTR system is crucial. The handwritten characters are more complex as compared to the printed characters due to the following reasons:

i. While writing, there exists a sort of clank in the collection process.
ii. An individual has diversified writing style triggering substantial divergence and character stroke variation.
iii. The situation and the condition of an individual on different occasions influences variation in his/her handwriting.
iv. Form and shape similarity.
v. Letters with a combination of matra and consonant clusters add additional intricacies. For these reasons, adopting a general classifier to classify the handwritten characters of several writers is not always a viable option.

Devanagari is a left-to-right alpha syllabary and it is originated from Brahmic script. It is mostly used in India, Tibet, Nepal, and South-East Asia [2]. It is used to write the literature or writings of Marathi, Hindi, Nepali, and other similar languages of East and South Asia. The Marathi language is a writing system which is adopted from Devanagari script, which comprises of 36 rudimentary consonants, 15 vowels, 2 sound modifiers (Chandrabindu and Halant), and 10 numeral characters. The Tables 1, 2 and 3 represents the basic consonant characters, vowel characters and digit characters respectively. Further, the individual consonant could be blended together with the vowels which forms 12 additional derived variants for individual consonant character. Table 4 depicts one example of this kind for 'ध'. Two different consonants can be combined together to form another special consonant cluster. Devanagari script also contains some special characters such as the two forms of raphars, that is the special consonant cluster formed by combining the consonant 'र' ('R') with other consonant. There is a special symbol for the sacred syllable om: ॐ. All these special consonant clusters and the sacred syllable are shown in Table 5. It also uses 17 punctuation marks, 8 reference marks and 2 diacritic marks.

Among all the datasets, DHCD is the best dataset for the rudimentary 36 consonants and 10-digit characters. It contains two thousand images for individual character hence, a total of 92 thousand images. It is very important to identify the consonants, consonant clusters, letters with combinations of matra, digits, and punctuation marks perfectly; as the effective meaning of the whole sentence gets changed if any one of the consonants, consonant clusters, letters with combinations with matra, digits, and punctuation marks are not recognized properly. There is one good example to demonstrate it in history. According to a legend, it is said that in August 1773, the Raghunath Rao the uncle of the Peshwa of Maratha empire Narayan Rao had sent a letter to Sumer Singh Gardi (the chief bodyguard of Peshwa family) to seize Peshwa Narayan Rao. The letter contained the message in Marathi as "नारायणरावांना धरा" (meaning that "Seize Narayan Rao"). But this letter was forged by Anandibai (wife of Raghunath Rao) and changed it to "नारायणरावांना मारा" meaning that kill Narayan Rao. It resulted in the death of Narayan Rao. There after a phrase came into in routine that is "'ध' चा 'मा' करणे" meaning "To change 'dha' to 'ma'" [3]. The same misunderstanding or some sort of conflict may come for the following sentence. "सकाळी दोन गोळ्या घेणे" meaning is ("Take two pills in the morning"), but it could be misinterpreted as "सकाळी दोन गोळ्या घालणे" meaning is ("shoot with two shots in the morning").

The classification task becomes more difficult in case of consonant clusters present in the handwritten text where first consonant forming the cluster is always present with its first half part in the cluster. It is also more important to identify the punctuation marks and their positions in the sentence otherwise the meaning of the sentence gets totally changed. For e.g. "मी चोरी करणार नाही, केल्यास मला शिक्षा करावी" meaning that "I will not steal, if I do, I should be punished". But if the position of comma is changed the meaning is totally different as: "मी चोरी करणार, नाही केल्यास मला शिक्षा करावी" meaning that "I will steal, if not, I should be punished" [4].

Segmentation and optical character recognition are the most important parts of any handwritten text recognition systems. It divides the handwritten text into several segments. The effectiveness of the handwritten system is dependent on how effectively the approach segments and identifies the digital output for handwritten texts. The strong is the image segmentation and OCR method more promising results can be obtained by the implemented system. The research work maintains a digitized document of the handwritten Devanagari script. Using transfer learning it will become simpler to train the model with additional number of classes for compound characters needed in Devanagari [5]. The training time taken by various classifiers is again a serious issue which is further required to be addressed. Various recognition systems are more expensive. It is observed that effective use of features to train the model results in less time required for training the model. If the model has low resource utilization in the feature extraction phase, then it makes the implemented system less expensive [6].

This approach implements an effective DHTR system. It uses the best suited methods required in various stages of DHTR. It configures the CNN in such a way that additional classes of Devanagari characters can be easily added to train the model. The training time is decreased due to the use of dropout and ReLU layers. The conflict resolution step of this approach results in less ambiguity in effectively recognizing the words. This approach is translation and rotation invariant. It has an advantage of low resource utilization in pre-processing, extracting features and training stages. It produces precise results irrespective of the size of the input image. It employs an effective segmentation technique that ideally segregates individual characters from the input word. The stages such as word reconstruction, dictionary searching and conflict resolution further helps in improving the accuracy of the proposed approach. The remainder of this paper is presented as: section 2 gives a brief overview of the recent researches related to the handwritten text retrieval. The pre-processing steps which are an integral part of the proposed approach are essential for segmentation are presented in section 3. Further, the section presents the implemented CNN transfer learning to enhance the text recognition. The section 4 reports the experimental results which reveals the effectiveness of the proposed work. Section 5 outlines the concluding remarks, limitations and future scope of the proposed work.

## 2. Related work

A huge amount of research is carried out in handwritten text recognition. The research in this domain was commenced in 1970. Devanagari text consists of basic, compound and numerical characters. The approaches developed mainly use the chain codes, statistical, structural, Vector Distance, zoning, gradient and Gaussian filter features and employs the various classifiers such as SVM, FFNN, fuzzy set, quadratic and minimum distance classifier. Some related studies in this regard are presented in this section.

In [7] the authors have segmented handwritten words into different segments by using line segmentation. It integrates the methods such as neighbor pixel analysis, projection and gap detection between text lines. The authors of [8] have proposed a projection profile-based algorithm which deals efficiently with skewed text and overlapped and touched lines. In [9], the authors have presented two methods for DHTR, the first that uses artificial neural network (ANN) with pattern recognition (PR) tool and the other uses the CNN. The authors also have compared both the methods. The work states that the CNN have provided best results over ANN with PR tool.

In [10], a method is proposed that recognizes 29 consonants and one modifier. The authors have created their own Devanagari script dataset which consists of 29 consonants without header line. It has 34604 handwritten images collected from different individuals. It employs Deep CNN for feature extraction and character recognition. The authors of [11] have suggested a character recognition system based on a set of primitives. It employs the handwritten strokes and speed at which the stroke is generated to recognize the Devanagari character. This approach presents a new feature termed as extended directional feature (EDF) set for the primitive identification. It uses strokes to plot the points on a letter. The number of points differs depending on the speed at which the stroke is written and the size of the stroke. The digitization device samples the data consistently over time and smoothens the points. The Discrete Wavelet Transform is used to compute the curvature points from the smoothened handwritten data which are used to recognize the character.

The authors of [12] have presented a three-step character recognition approach. The first step is image correction, where it uses the principal component analysis (PCA) to find out the hidden part and get the information about the complex data present in the image. This step is achieved by gray scaling, binarization, and noise removal. The second step is Segmentation. This step uses line, word, fused word, and character segmentation for the segmentation. The final step of this approach is recognition where it employs Eigen space algorithm for the character recognition. In work [13], the authors have employed segmentation, control points computation of a Bezier and k-NN classifier methods for leaf classification. In work [2], a deep learning architecture is proposed to extricate the characters in DHTR. The DHCD comprised of 92 thousand images of 36 Devanagari consonant and 12 vowel characters is used for training the model. It employs the CNN for extracting the feature and classification. The authors of [6] have proposed a method for leaf classification that employs segmentation for dividing the leaf image into 10 segments. The method further calculates the control points of a Bezier of leaf boundary and Fourier descriptor values and selected points from curve boundary as a feature

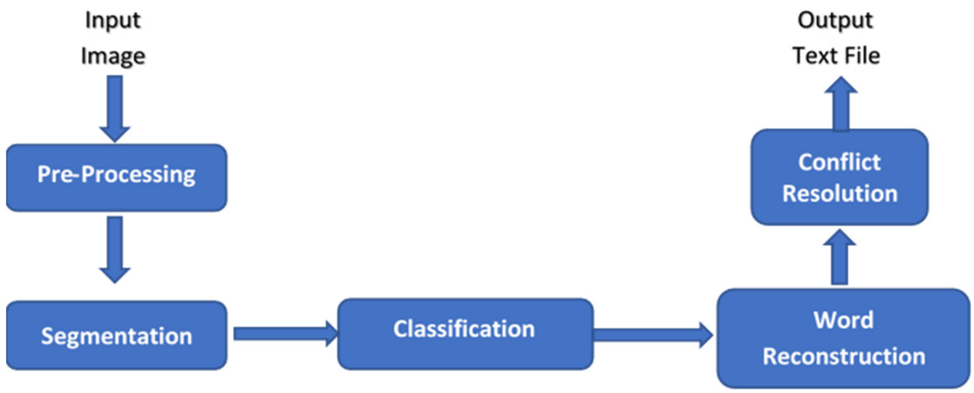


**Fig. 1.** Algorithmic flow of the proposed method.

for leaf classification. For classification the CapsNet is configured for the novel feature vector.

In the method [14] authors have proposed a method for classification of handwriting Devanagari text using CNN. The implemented CNN has five convolutional and 3 fully connected layers where last two fully connected layers has dropout layers. In [15], the authors have presented a detailed survey of a hot new architecture termed as CapsNet which was initially proposed for handwritten digit classification. The architecture is based on the concepts of inverse graphics and produces promising results in image processing tasks. The authors of [16] have outlined a SVM Hindi handwritten text recognition system where, it initially separates the characters from word in preprocessing stage. In feature extraction k-means clustering is employed. Finally, the classification stage uses SVM for recognition of the characters.

In [17], a Hindi OCR is presented that efficiently performs feature extraction and achieves classification with multiple classifiers. The researchers in [18], carries out the preprocessing stage consisting of binarization, statistical noise reduction, skew detection, followed by segmentation and thinning. Further, the features are extracted, and classification is done through numerous classifiers viz. Rough Fuzzy MLP, Fuzzy Rough SVM, Fuzzy Markov Random Fields and Fuzzy SVM. The authors of [19], have presented a Hindi OCR using binarization, Shiro Rekha removal, K-means clustering, and linear kernel based SVM for Hindi handwritten text classification. In work [20], an offline Devanagari OCR method is outlined that classifies the characters by extracting the strength, angle and histogram of gradient (SOG, AOG, HOG) based directional features. It adopts a 3-fold cross validation based combined classifier for character classification. Devanagari OCR method in [21] extracted chain coding features using PCA and LDA and it used gradient and directional features for edge recognition. The approach employs SVM for classification. The various architectures used in the literature has one major drawback of conflicts due to character similarities. Therefore, sometimes the recognized word is different hence a conflict resolution is essential.

The authors of [22] have implemented a two-stage VGG16 architecture for DHTR with adaptive gradient methods. In [23], a self-made dataset made up of 29 consonants and one modifier is used for DHTR. It uses deep CNN for classification of the handwritten Devanagari characters. SVM is employed in the work [24] for classification of handwritten and printed mono-lingual documents in Marathi, Sanskrit and Hindi languages. The research in [25] uses ANN whereas the work in [26] and [27] uses the CNN for DHTR. These studies lag in conflict resolution and poor selection of hidden layers such as the dropout layer resulting in poor performance.

By keeping in mind, the shortcomings of various approaches in DHTR, this work implements an effective and efficient DHTR system using various latest techniques for pre-processing, segmentation and recognition. The qualities like conflict resolution are provided for yet less ambiguity. The methodologies used and overall implementation details of this work are discussed in upcoming sections.

## 3. Methodologies

The proposed research work consists of five major components viz. Pre-processing, Segmentation, Character Recognition, Word Reconstruction, and Conflict Resolution. The following Fig. 1 depicts the stepwise representation of this work. It supports input images of any format.

### 3.1. Pre-processing

The image of the document will be taken as an input query image. Initially, the pre-processing step is applied which includes identifying borders, cropping the borders, applying transformation to straighten the page, removing noise, correcting skew, size normalization and applying sharpening kernel. The initial input image is scanned and analyzed to prepare it for future processing. The Live Corp tool is used for border checking, cropping and straightening the page. After detecting the border of given page, the noise is removed using Gaussian filter with Otsu threshold. Further, the image is converted into grey scaled image [28–30].

### 3.2. Character segmentation

The next major step after image preprocessing is segmentation of the image into words. The Tessaract-ocr an open-source library [31], is used for segmentation, that outputs a stream of characters that will be passed on to the next stage. The next stage performs the character segmentation by dividing the input into 3 phases: upper, middle and lower zones. The middle zone represents the actual word. This phase helps in checking whether words are joined or not. This phase corps the characters to convert them into a 32 x 32 sized image.

### 3.3. Recognition

The classification stage will classify the characters into different Devanagari characters that are available in the Dataset used for training i.e., DHCD. After the words have been scanned, the written or printed character is compared with the similar character stored within the predefined class, for the classification of every character. This stage uses CNN for classification with 46 classes from Devanagari script. Each character is therefore classified from the cropped image [2].

A CNN is a multi-layered neural network with a specific architecture to identify complex features in data. The following Fig. 2

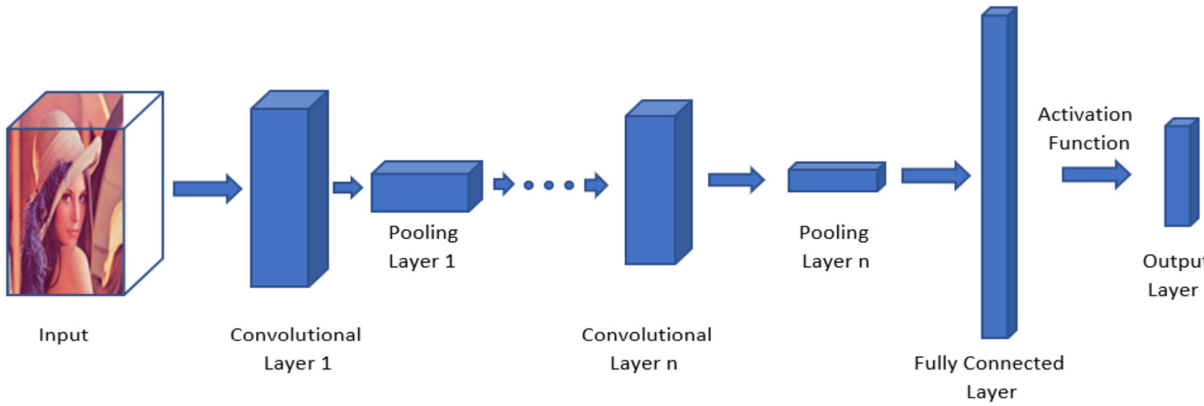


**Fig. 2.** General CNN architecture.

**Table 6**
Configurations of runtime environments where the proposed approach is implemented.

| Sr. No. | System | Configuration | | | | |
|---|---|---|---|---|---|---|
| | | Processor | GPU | Clock Speed (GHz) | RAM (GB) | OS |
| 1 | Laptop #1 | Core i5 | - | 2.50 | 8 | Ubuntu 18.04 LTS |
| 2 | Laptop #2 | Core i7 | - | 2.50 | 8 | Ubuntu 18.04 LTS |
| 3 | Laptop #3 | Core i7 | GeForce GTX 860 m 4GB | 2.50 | 8 | Windows 10 |

shows the architecture of a general CNN. It can be applied to classify the contents of various images. The images can be provided as an input into the model. Like ANN, CNN is also influenced by the workings of the human brain. CNNs are able to classify images by extracting features, much like how the human brain searches for features to identify the objects. The CNN has set of convolutional layers and maxpooling layers. The nth pooling layer is connected to fully connected layer. It carries few backpropagation steps in learning phase in order minimize the loss. It finally uses some activation function such as Softmax or Tanh to generate the output [32,33].

### 3.4. Word reconstruction

Dictionary searching and Conflict resolution: The prediction of word will be based on the reference from the dictionary. Once the character is predicted to something, it will get some confidence score. Instead of having the best prediction as the output we will take the top 2 predictions as final output but, if a single class has confidence over 60%, then it is taken as a final output. After all the classes are found for a single word, all the possibility will be checked for those classes in dictionary. The best fit will be chosen as the final word. Now, if there is any ambiguity in the word a final edit option will be provided to the user to make changes as per his/her convenience. In this phase, with the help of given dictionary, words are mapped. It is used to check whether the word is correct or not. It also gives suggestions if the word is similar to 3-4 words. The data can be edited dynamically. The final output of this work is the recognized text in an output file in the text format [34]. The following Fig. 3 shows the complete flowchart of the proposed method.

## 4. Experimental setup and results

Python 3.7, Tensorflow GPU [35] library and Keras software [36] are used to implement the proposed work. Several experiments are carried out on various systems. The configuration of the systems which are used for the experiments is described in Table 6.

### 4.1. Proposed CNN architecture

The implemented CNN has 4 convolutional, 2 maxpooling, 3 drop out, a flatten and 2 dense layers. The following Fig. 4 depicts the parameter count associated with each layer. The filter size considered here is 3 x 3. The model has total 486670 parameters. It has been configured for classifying the input text into 46 different classes. The architecture of the proposed CNN is depicted in Fig. 4 and Fig. 5 shows the summary of the same.

### 4.2. Evaluation parameters

The most commonly used evaluation parameter termed as accuracy is employed to evaluate the effectiveness of the proposed method [6]. It could be expressed by:

$$\text{Accuracy} = \frac{\text{Count of accurately classified images}}{\text{Count of testing images}} * 100 \tag{1}$$

The proposed method is compared in terms of accuracy with several state-of-the-art DHTR methods, recognized as D. Khanduja et al. [37], A. Shinde et al. [38], S. Puri et al. [16] and S. Acharya et al. [2]. The results are represented in Table 7. It is noticeable that the proposed approach gives high accuracy. Furthermore, it outperforms in processing time. The training time taken by proposed method is represented in Table 8. Various researches have proved that the neural networks can be trained on GPUs in reduced time [39], therefore, for analyzing the same, the proposed method is implemented on both GPUs and CPUs. It is noted from the results that the training time required on CPU is almost 10 times larger than GPU. 16.6% of all the input images are reserved as test samples for testing the model.

The following Fig. 6 represented the model's training and testing accuracy obtained against number of epochs. It has been observed that after 5 to 8 epochs the accuracy of the model increased tremendously. After 27 epochs the accuracy figures raise steadily. Fig. 7 shows the graph of model's training and testing loss against epochs. It has been observed that after ten epochs the loss figures reduced substantially.

The following Fig. 8 shows a sample word image which is input to the proposed approach. It is clearly visible from the output that

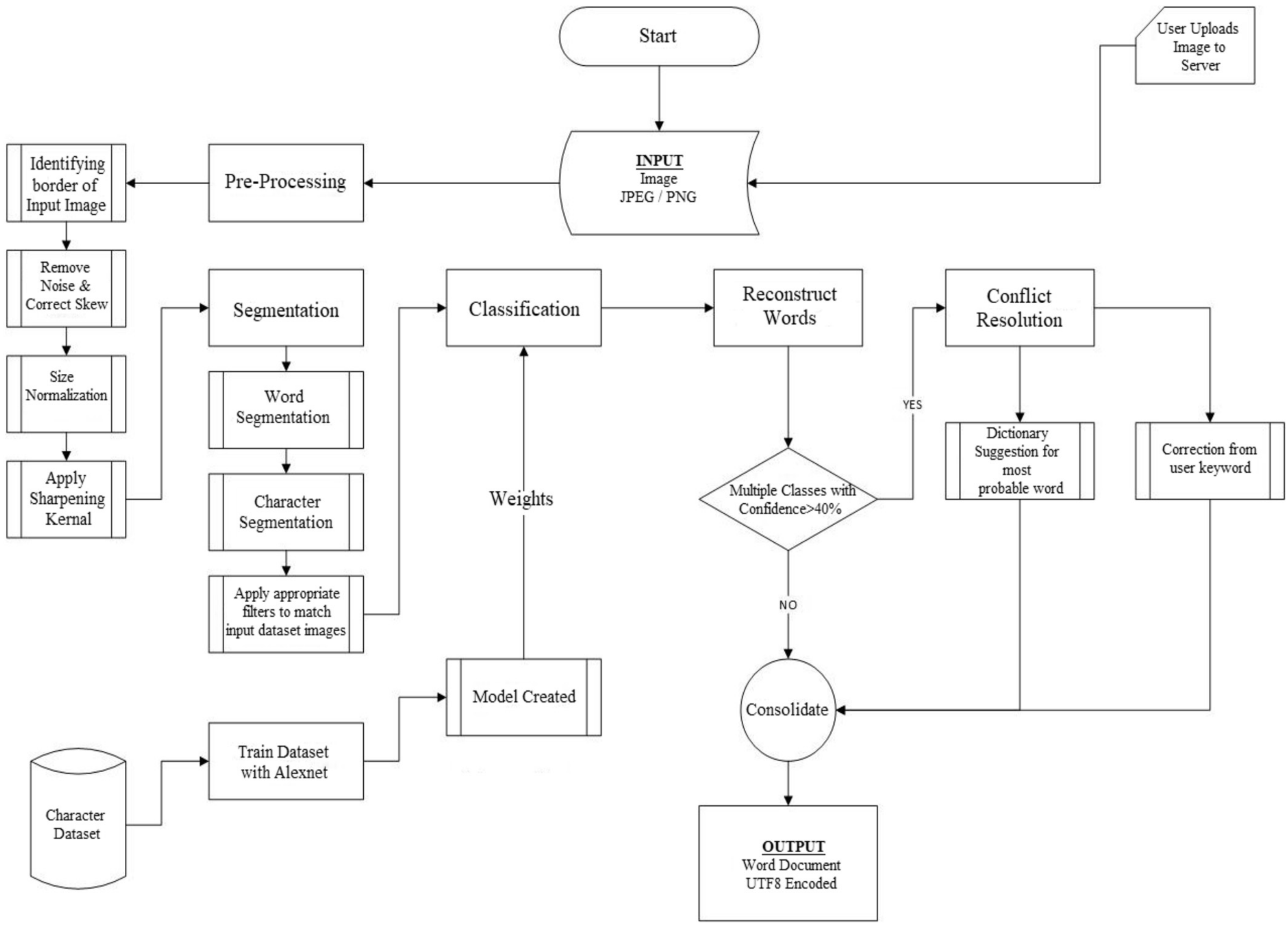


**Fig. 3.** Detailed flowchart of proposed approach.

**Table 7**
Results.

| Authors | Methods used | Number of classes | Accuracy (%) |
|---|---|---|---|
| Deepti Khanduja et al. [37] | MLP | 10 | 93.4 |
| Ambadas Shinde et al. [38] | 20 Layer CNN | 104 | 98 |
| Shalini Puri et al. [16] | SVM Classifier | 51 | 98.35 |
| Shailesh Acharya et al. [2] | CNN | 46 | 98.47 |
| Proposed Approach | 12 Layer CNN including Dropout Layer | 46 | 99.13 |

**Table 8**
Training time for CPU and GPGPU.

| Sr. No. | System configuration | Training time (Seconds) |
|---|---|---|
| 1 | Laptop #1 | 1492.7929 |
| 2 | Laptop #2 | 1121.5071 |
| 3 | Laptop #3 | 157.9840 |

the identified word properly segmented. The system classified it as and the Confidence score of 97.67% is produced for the same.

## 5. Conclusion

As loads of handwritten documents are produced nowadays in various government and educational organizations, quality handwritten character recognition systems are critical. The proposed approach implements a DHTR system that segments the input handwritten Devanagari text query image, isolates individual characters and assesses the resemblance of the individual character in its repository with the isolated characters of the query image within an anticipated range of resemblance. It is observed that the proposed system yields better performance. The proposed work picks statistical features that are computed from isolated characters from the input query image. For the character class repository of limited diversity, which includes very few homogeneous patterns is used. The proposed approach outperforms as compared to the other handwritten character recognition systems in terms of performance and accuracy. The conflict resolution step results in less ambiguity in recognition of words. The authors of the work hope that the end-user may get speedier and reliable handwritten recognition system to digitize his/her valuable documents. The digitization process facilitates the long-term storage of data, reduction in the possibilities of manipulation of data, and efficient searching of the data. It further prevents the data from natural calamities such as floods, fires, tsunamis, and earthquakes and eating the papers by pests. The work finds its limitation in its abundant dependence on database for recognition. In future, the system's accuracy and reliability can be further improved by adding new classes of consonant clusters into the dataset. Recently, G. E. Hinton have

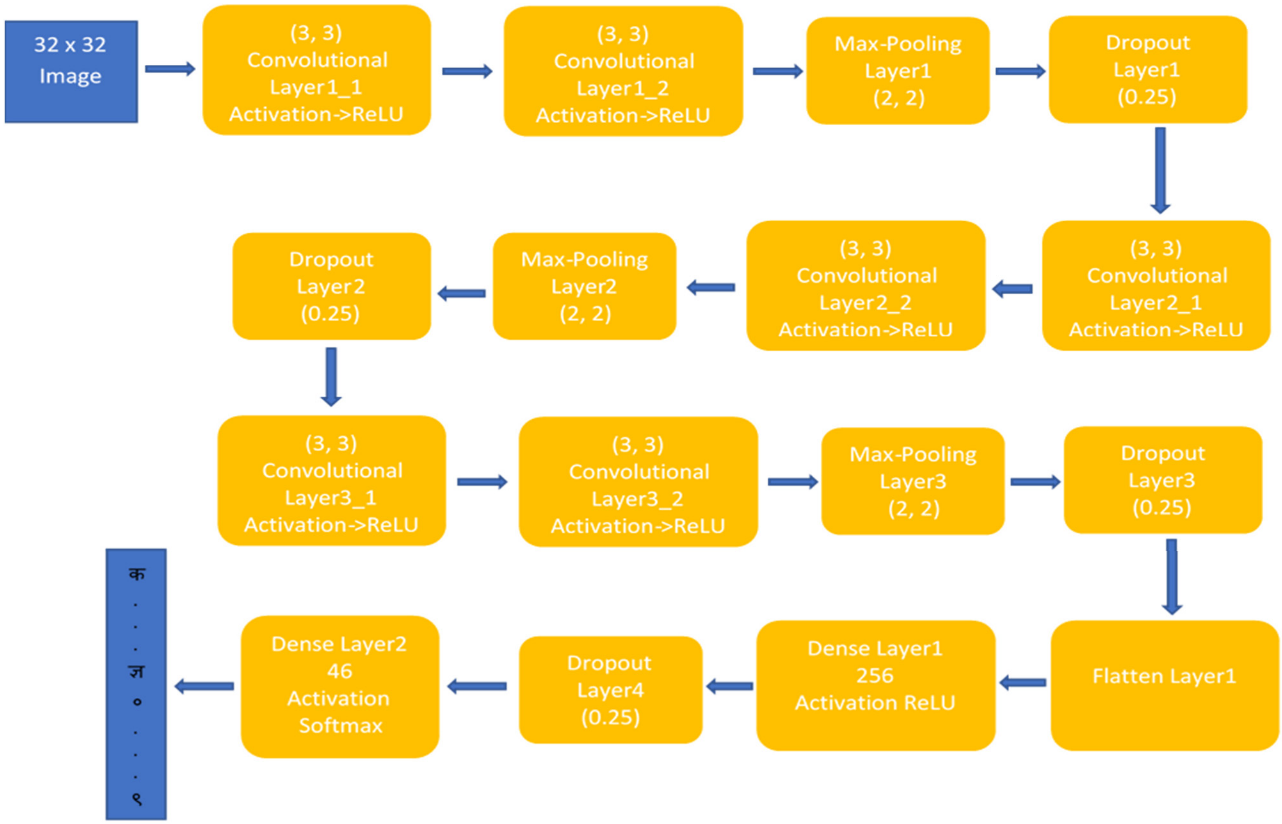


**Fig. 4.** Proposed CNN architecture.

```
model.summary()

Layer (type)                 Output Shape              Param #
=================================================================
conv2d_1 (Conv2D)            (None, 30, 30, 32)        320
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 28, 28, 32)        9248
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 14, 14, 32)        0
_________________________________________________________________
dropout_1 (Dropout)          (None, 14, 14, 32)        0
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 12, 12, 64)        18496
_________________________________________________________________
conv2d_4 (Conv2D)            (None, 10, 10, 64)        36928
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 5, 5, 64)          0
_________________________________________________________________
dropout_2 (Dropout)          (None, 5, 5, 64)          0
_________________________________________________________________
flatten_1 (Flatten)          (None, 1600)              0
_________________________________________________________________
dense_1 (Dense)              (None, 256)               409856
_________________________________________________________________
dropout_3 (Dropout)          (None, 256)               0
_________________________________________________________________
dense_2 (Dense)              (None, 46)                11822
=================================================================
Total params: 486,670
Trainable params: 486,670
Non-trainable params: 0
_________________________________________________________________
```

**Fig. 5.** The summary of the proposed CNN model.

proposed a new concept known as GLOM [40] based on agglomeration which is just an idea and not mechanized yet. This concept if implemented, may transform the whole deep learning domain in surprisingly unpredictable manner. In future, this concept may be used to overcome the issue of incapability to retrieve consonant clusters and highly identical consonants. This work finds its applications in schools, colleges, offices like courtrooms, police stations, revenue offices, Municipal Corporation offices, Government hospi-

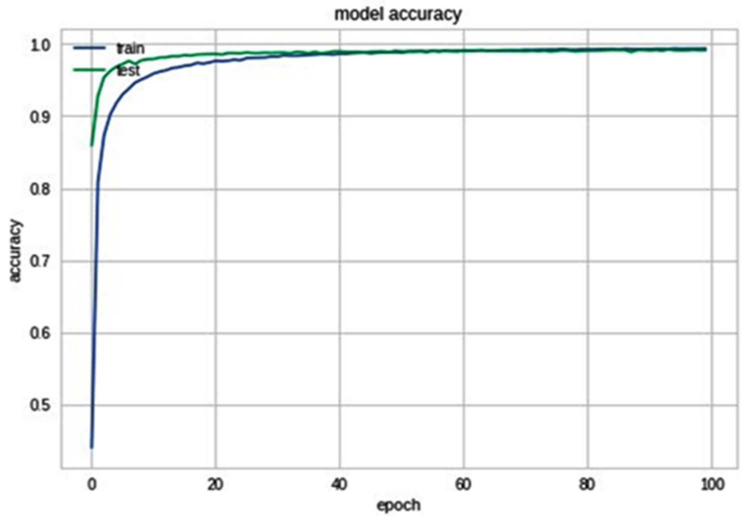


**Fig. 6.** Proposed model's graph of training and testing accuracy against number of epochs.

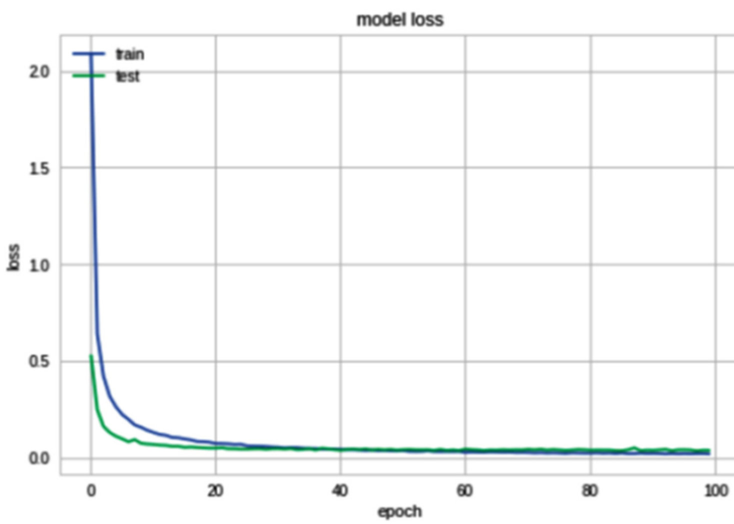


**Fig. 7.** Proposed model's graph of training and testing loss against number of epochs.

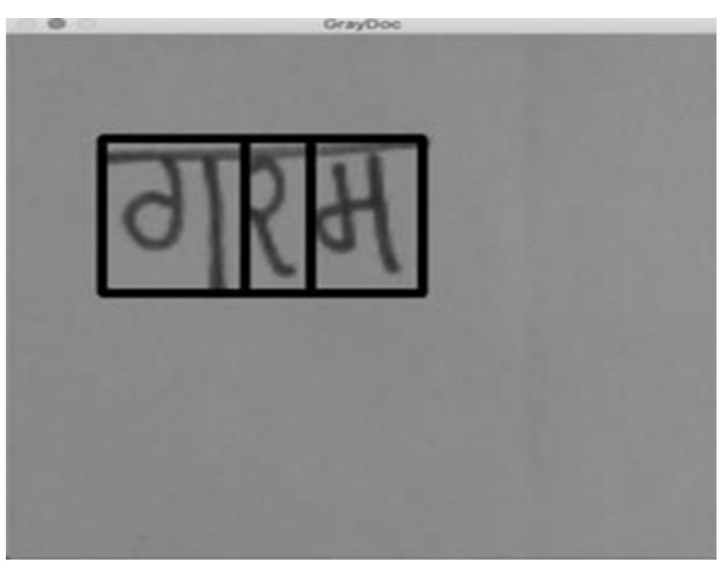


**Fig. 8.** Sample input.

tals, etc. It can also be used for blind people as they will be able to understand the handwritten documents by using the added voice features. It can be further useful for developing a treatment for handling mentally retarded patients by generating knowledge from digitization of ancient Vedic literature and medical prescriptions.

## Declaration of competing interest

No competing interest.

## References

[1] R.H. Singh, K. Narsimhamurthy, G. Singh, Neuronutrient impact of Ayurvedic Rasayana therapy in brain aging, Biogerontology 9 (6) (2008) 369–374.
[2] S. Acharya, A.K. Pant, P.K. Gyawali, Deep learning based large scale handwritten Devanagari character recognition, in: 2015 9th International Conference on Software, Knowledge, Information Management and Applications (SKIMA), 2015, pp. 1–6.
[3] https://ancientales.wordpress.com/2017/03/04/the-forsaken-cries-of-peshwa-narayan-rao/.
[4] https://mr.wikisource.org/wiki/%E0%A4%AA%E0%A4%BE%E0%A4%A8:%E0%A4%A4%E0%A4%B0%E0%A5%8D%E0%A4%95%E0%A4%B6%E0%A4%BE%E0%A4%B8%E0%A5%8D%E0%A4%A4%E0%A5%8D%E0%A4%B0.pdf/231.
[5] N. Sahu, M. Sahai, Segmentation of Devanagari character, in: 2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom), 2016, pp. 4005–4008.
[6] SandeepPande, Manna Sheela Rani Chetty, Bezier curve based medicinal leaf classification using capsule network, Int. J. Adv. Trends Comput. Sci. Eng. 8 (6) (2019) 2735–2742.
[7] S. Goyal, A. Bhatia, Method for line segmentation in handwritten documents with touching and broken parts in Devanagari script, Int. J. Comput. Appl. 102 (2014) 22–27.
[8] R. Garg, N.K. Garg, An algorithm for text line segmentation in handwritten skewed and overlapped Devanagari script, Int. J. Emerg. Trends Eng. Develop. 4 (5) (2014) 114–118.
[9] S. Duddela, R. Mehta, Nisha Sharma, Classification of handwritten Devanagari number – an analysis of pattern recognition tool using neural network and CNN, Proc. Comput. Sci. 167 (2020) 2445–2457.
[10] M. Jangid, S. Srivastava, Handwritten Devanagari character recognition using layer-wise training of deep convolutional neural networks and adaptive gradient methods, J. Imaging 4 (2) (2018).
[11] V.L. Lajish, S.K. Kopparapu, Online handwritten Devanagari stroke recognition using extended directional features, in: 8th International Conference on Signal Processing and Communication Systems (ICSPCS), 2014, pp. 1–5.
[12] A. Deshmukh, R. Meshram, S. Kendre, K. Shah, Handwritten Devanagari character recognition, Int. J. Eng. Res. Technol. 03 (04) (April 2014) 842–846.
[13] Sandeep Pande, Manna Sheela Rani Chetty, Position invariant spline curve based image retrieval using control points, Int. J. Intell. Eng. Syst. 12 (4) (2019) 177–191.
[14] P.K. Sonawane, S. Shelke, Handwritten Devanagari character classification using deep learning, in: International Conference on Information, Communication, Engineering and Technology (ICICET), Pune, 2018, pp. 1–4.
[15] Sandeep Pande, Manna Sheela Rani Chetty, Analysis of capsule network (capsnet) architectures and applications, J. Adv. Res. Dyn. Control Syst. 10 (10) (2018) 2765–2771.
[16] S. Puri, S.P. Singh, An efficient Devanagari character classification in printed and handwritten documents using SVM, Proc. Comput. Sci. 152 (2019) 111–121.
[17] M. Yadav, R. Purwar, Hindi handwritten character recognition using multiple classifiers, in: Seventh International Conference on Cloud Computing, Data Science & Engineering – Confluence, 2017, pp. 149–154.
[18] A. Chaudhuri, K. Mandaviya, P. Badelia, S.K. Ghosh, Optical character recognition systems for Hindi language, in: Optical Character Recognition Systems for Different Languages with Soft Computing: Studies in Fuzziness and Soft Computing, vol. 352, 2017, pp. 193–216.
[19] A. Gaur, S. Yadav, Handwritten Hindi character recognition using K-means clustering and SVM, in: Fourth International Symposium on Emerging Trends and Technologies in Libraries and Information Services, 2015, pp. 65–70.
[20] M. Bhalerao, S. Bonde, A. Nandedkar, S. Pilawan, Combined classifier approach for offline handwritten Devanagari character recognition using multiple features, in: D. Hemanth, S. Smys (Eds.), Computational Vision and Bio Inspired Computing, in: Lecture Notes in Computational Vision and Biomechanics, vol. 28, 2018, pp. 45–54.
[21] S. Shitole, S. Jadhav, Recognition of handwritten Devanagari characters using linear discriminant analysis, in: Second International Conference on Inventive Systems and Control, 2018, pp. 100–103.
[22] S.P. Deore, A. Pravin, Devanagari handwritten character recognition using fine-tuned deep convolutional neural network on trivial dataset, Sadhana 45 (243) (2020).
[23] Y. Gurav, P. Bhagat, R. Jadhav, S. Sinha, Devanagari handwritten character recognition using convolutional neural networks, in: 2020 International Conference on Electrical, Communication, and Computer Engineering (ICECCE), 2020, pp. 1–6.
[24] S. Puri, S.P. Singh, An efficient Devanagari character classification in printed and handwritten documents using SVM, Proc. Comput. Sci. 152 (2019) 111–121.
[25] P.B. Khanale, S.D. Chitnis, Handwritten Devanagari character recognition using artificial neural network, J. Artif. Intell. 4 (2011) 55–62.
[26] R. Guha, N. Das, M. Kundu, M. Nasipuri, K.C. Santosh, DevNet: an efficient CNN architecture for handwritten Devanagari character recognition, Int. J. Pattern Recognit. Artif. Intell. 34 (12) (2020) 2052009.
[27] S. Susan, J. Malhotra, Recognising Devanagari script by deep structure learning of image quadrants, J. Libr. Inform. Technol. 40 (5) (Sept 2020) 268–271.
[28] E. Hussein, Computed Radiation Imaging, Elsevier, ISBN 9780123877772, 2011, pp. 97–123.
[29] https://likegeeks.com/python-image-processing/.
[30] M. Sonka, V. Hlavac, R. Boyle, Image processing, analysis, and machine vision, in: Global Engineering: Timothy L. Anderson, 4th edition, ISBN 9789386858146, 2015.
[31] https://nanonets.com/blog/ocr-with-tesseract/.
[32] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, Proc. IEEE 86 (11) (1998) 2278–2324.
[33] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, in: Advances in Neural Information Processing Systems, 2012, pp. 1097–1105.
[34] I. Kissos, N. Dershowitz, OCR error correction using character correction and feature-based word classification, in: 2016 12th IAPR Workshop on Document Analysis Systems (DAS), 2016, pp. 198–203.
[35] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, et al., Tensorflow: a system for large-scale machine learning, in: Proc. 12th USENIX Symposium on Operating Systems Design and Implementation, Savannah, GA, USA, 2016, pp. 265–283.
[36] F. Chollet & others, Keras, Available at: https://github.com/fchollet/keras, 2015.
[37] D. Khanduja, N. Nain, Subhash Panwar, A hybrid feature extraction algorithm for Devanagari script, ACM Trans. Asian Low-Resour. Lang. Inf. Process. 15 (1) (2015) 2.
[38] A. Shinde, Y. Dandawate, Convolutional neural network based handwritten Marathi text recognition, J. Xidian Univ. 14 (8) (2020) 528–536.
[39] Sandeep Pande, Manna Chetty, Fast medicinal leaf retrieval using CapsNet, in: S. Bhattacharyya, J. Nayak, K.B. Prakash, B. Naik, A. Abraham (Eds.), International Conference on Intelligent and Smart Computing in Data Analytics, in: Advances in Intelligent Systems and Computing, vol. 1312, Springer, Singapore, February 2021, pp. 149–155.
[40] G.E. Hinton, How to represent part-whole hierarchies in a neural network, ArXiv, arXiv:2102.12627 [abs], 2021.