Nepal Script Text Recognition using CRNN CTC Architecture

Swornim Nakarmi, Sarin Sthapit, Arya Shakya, Rajani Chulyadyo, Bal Krishna Bal
Kathmandu University
Dhulikhel, Nepal
{sn34021319, ss53021319, as48021319}@student.ku.edu.np
{rajani.chulyadyo, bal}@ku.edu.np

Abstract

Nepal Script (also known as Prachalit Script) is the widely used script of Nepal Bhasa, the native language of the Kathmandu Valley in Nepal. Derived from the Brahmi Script, the Nepal Script was developed in the 9^{th} century and was extensively used till the 20^{th} century, before being replaced by the Devanagari script. Numerous ancient manuscripts, inscriptions, and documents written in the Nepal Script are still available containing immense knowledge on architecture, arts, astrology, ayurveda, literature, music, tantrism, etc. To preserve and revive Nepal Bhasa, digitizing such documents plays a crucial role. This paper presents our work on text recognition for the Nepal Script. The implementation includes the Nepal Script text recognizer based on CRNN CTC architecture aided by line and word segmentations. Leveraging a carefully curated dataset that encompasses handwritten and printed texts in the Nepal Script, our work has achieved CER of 6.65% and WER of 13.11%. The dataset used for this work is available as Nepal Script Text Dataset on Kaggle. The paper further explores the associated challenges due to the complex nature of the script such as conjuncts, modifiers and variations; and the current state of the script.

Keywords: Nepal Bhasa, Nepal Script, Under-resourced, Off-line Text Recognition, CRNN CTC

1. Introduction

Enabling a computer system to recognize handwritten as well as printed texts in any script is essential to convert such texts into digital and editable form. Such systems have been developed for some of the widely used scripts like Roman, Devanagari, etc., with high accuracy. However, this is not true in case of the other regional scripts. Among such scripts is the Nepal Script (also known as Nepal Lipi, Prachalit Nepal Script), which was widely used in ancient Nepal for writing Nepal Bhasa, the native language of the Kathmandu Valley in Nepal. Unfortunately, Nepal Bhasa along with the Nepal Script were marginalized in modern Nepal due to political influence. The Nepal Script can be found on many ancient manuscripts, inscriptions, scriptures, artifacts, and other forms of writing. Such documents reflect an important aspect of history and tradition, and, therefore, need to be preserved, and thus digitised.

However, the digitization of the Nepal Script is hindered because it is under-resourced, owing to a lack of funding, dedicated research, and technological infrastructure. As a result, comprehensive datasets are not available for developing such text recognition systems. Additionally, unlike in ancient times when the Nepal Script was used widely for various purposes, its usage has declined significantly over the years. Mainly due to the dominance of other scripts, particularly Devanagari Script, this has led to the decreased relevance of the Nepal Script in contemporary society. These challenges highlight the need for more attention and resources to be allocated towards the preservation and digitization of this aspect of Nepal's cultural heritage.

Text recognition is a challenging research area where the intricacies of the scripts complicate the text detection process. While the Roman Script presents relatively simpler shapes and benefits from the widespread availability of resources, dedicated research and datasets, the Devanagari Script introduces additional complexity due to its more intricate characters and rules. This means that tailored approaches and innovative techniques are needed to effectively address the diverse demands of text recognition across various scripts and languages. Additionally, the complexities inherent in the Nepal Script, characterized by intricate character shapes, historical variations, and limited available resources, pose significant challenges in text recognition, necessitating specialized approaches and dedicated efforts for accurate and efficient recognition systems.

The performance of text recognition systems have excelled recently due to the emergence and advancements of Deep Learning techniques. The integration of Deep Learning methods with text detection has enhanced the capabilities of text recognition systems, enabling them to handle diverse scripts and languages with greater accuracy and efficiency. The extracted text can be stored, edited and distributed more efficiently and effectively, facilitating tasks such as historical document preservation, healthcare data

SIGUL2024 Workshop, pages 244–251
21-22 May, 2024.

management and beyond.

In this paper, we propose a text recognition system based on a Deep Learning technique for recognizing texts written in the Nepal Script. To the best of our knowledge, our system is the first of its kind at the time of the writing of the paper. One of the key contributions of our work lies in the development of a tailored framework that addresses the intricacies and complexities of the Nepal Script, effectively overcoming obstacles such as complex character shapes, historical evolution of the entire script, adaptation to multilingual contexts, and limited available resources.

By leveraging the capabilities of Deep Learning techniques, our system demonstrates remarkable accuracy and efficiency in recognizing texts written in this script. Furthermore, we provide a comprehensive discussion on the dataset developed in our study, shedding light on its composition, size, and relevance to the task of the Nepal Script text recognition, which we plan to publish along with this paper. The paper not only presents a pioneering text recognition system for the Nepal Script but also offers valuable insights into the challenges and opportunities inherent in this endeavor. We anticipate that our contributions will inspire further exploration and advancements in the field of text recognition for underrepresented and under-resourced scripts like the Nepal Script.

2. Background

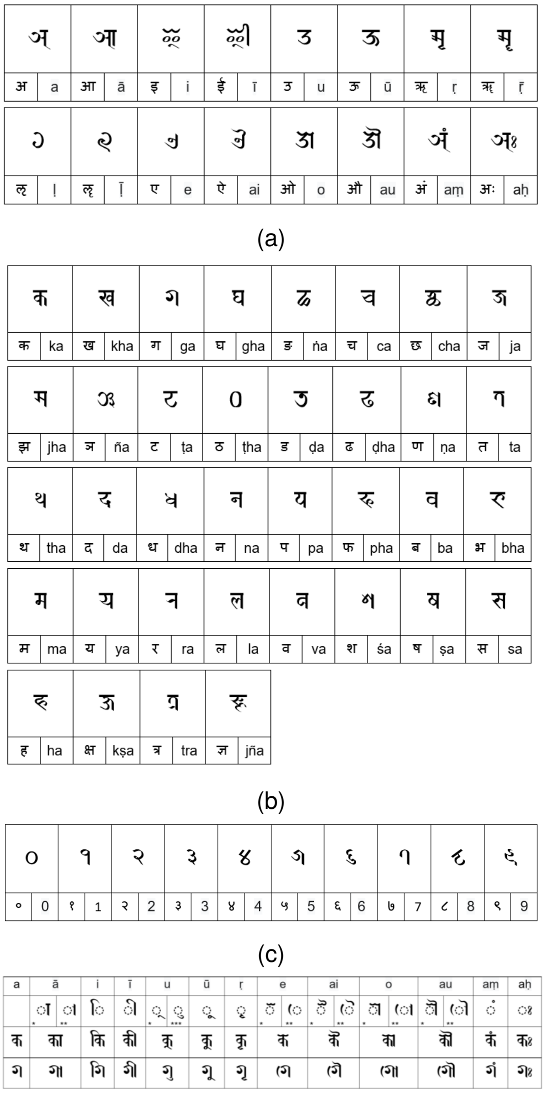
Derived from the Brahmi Script, the Nepal Script was developed in the 9^{th} century, and was prevalent till the 20^{th} century (Nepal Lipi Guthi, 1992). The earliest recorded manuscript written in the script is *Laṅkāvatāra Sutra* (908 AD) (Tamot, 1991). Other scripts such as *Ranjanā*, *Bhujiṃmol*, *Golmol*, *Litumol*, *Pācumol*, *Kveṃmol*, *Hiṃmol*, and *Kuṃmol* originated from this particular script. Apart from Nepal Bhasa, this script has also been employed for religious purposes and literature to transcribe Sanskrit, Pali, Maithili, and Bengali. Many century-old manuscripts, inscriptions, and documents scribed in Nepal Script endure, preserving extensive knowledge spanning arts, architecture, ayurveda, astrology, literature, music, and more.

Although the Nepal Script was extensively employed in the past, it experienced a significant decline primarily due to political factors, leading to its replacement by Devanagari Script for several decades. However, the recent efforts focused on advocacy and awareness have led to its resurgence, accompanied by a surge in its user base. It is worth noting that the script has been recently incorporated into the local curricula of various governmental bodies within the Kathmandu Valley. The current users of the Nepal Script encompass a diverse range of individuals, including Nepal Bhasa speakers, script enthusiasts, scholars, and students. Additionally, the development of numerous tools, applications, and fonts, alongside the recent introduction of its Unicode standard (Unicode, Inc., 2023), has facilitated its adoption across a wide range of devices. Following the release of the Unicode for the Nepal Script, its accessibility and usage have expanded significantly through digital platforms and media. The script is primarily used for Nepal Bhasa, the indigenous language of the Kathmandu Valley. However, it is also used for writing religious texts in languages such as Sanskrit and Pali.

Having originated during the same era, the Nepal Script shares numerous similarities with Devanagari and Bangla Scripts. For example, the presence of a header line (*śirorekhā* or *mvaḥ*) and the division of characters into upper, middle, and lower parts are common in all these three scripts.

The Nepal Script comprises of 16 vowel letters, 36 consonant letters, and 10 numerals, supplemented by an array of conjuncts and special symbols. Vowels, consonants, numerals, and modifiers are shown in Figure 1a, 1b, 1c, and 1d respectively. The presence or absence of the header line determines the way in which certain vowel modifiers are used. Additionally, there are numerous possible conjuncts, variations in characters and structure of characters during conjunct formation and vowel modifier usage. Every consonant has a distinct point to use *ukār* and *ṛkār*; and to join other consonants in a conjunct, referred to as *mhutupvāḥ*.

Considering the success of text recognition systems for similar scripts like Devanagari and Bangla Scripts, there is a compelling motivation to develop an offline text recognition system for the Nepal Script. The intricacies of the Nepal Script, characterized by complex shapes and similar-looking characters pose significant challenges for text recognition. Effective text recognition always requires a large and robust dataset, which is lacking for the Nepal Script. To resolve this, we prepared a dataset comprising handwritten and printed texts in the Nepal Script. The dataset includes a wide range of characters, conjuncts, modifiers and special symbols. As deep learning techniques demand a large dataset, a common practice to increase the size of image datasets is to apply various data augmentation methods (Shorten and Khoshgoftaar, 2019). Applying such techniques, we could augment our dataset, which is then fed to our model that utilizes the CRNN CTC architecture (Shi et al., 2016), shown in Figure 2.



अ	a	आ	ā	इ	i	ई	ī	उ	u	ऊ	ū	ऋ	ṛ	ॠ	ṝ
ऌ	ḷ	ॡ	ḹ	ए	e	ऐ	ai	ओ	o	औ	au	अं	aṃ	अः	aḥ

(a)

क	ka	ख	kha	ग	ga	घ	gha	ङ	ṅa	च	ca	छ	cha	ज	ja
झ	jha	ञ	ña	ट	ṭa	ठ	ṭha	ड	ḍa	ढ	ḍha	ण	ṇa	त	ta
थ	tha	द	da	ध	dha	न	na	प	pa	फ	pha	ब	ba	भ	bha
म	ma	य	ya	र	ra	ल	la	व	va	श	śa	ष	ṣa	स	sa
ह	ha	क्ष	kṣa	त्र	tra	ज्ञ	jña								

(b)

०	0	१	1	२	2	३	3	४	4	५	5	६	6	७	7	८	8	९	9

(c)

a ā i ī u ū ṛ e ai o au aṃ aḥ
Consonants
* Used for consonants with headerline
** Used for consonants without headerline
*** Used for ञ, ग, ए, श

(d)

Figure 1: (a) Vowels, (b) Consonants, (c) Numerals, (d) Modifiers used in Nepal Script with their corresponding character in Devanagari Script and transliterated form.

With this, we aim to develop a robust and accurate text recognition system for the Nepal Script, which is capable of safeguarding the linguistic heritage and cultural legacy of Nepal for future generations.

3. Related Works

Although Nepal script and Devanagari script are similar, less research has been done on text recognition in the former than in the latter. For Devanagari and Indic scripts, benchmark handwritten character databases are accessible to the general public. The OCR community makes extensive use of both machine learning (Shaw et al., 2008, 2014; Singh et al., 2011; Pant and Bal, 2016), and deep learning techniques (Dutta et al., 2018; Dwivedi et al., 2020; Mondal and Jawahar, 2022; Acharya et al., 2015).

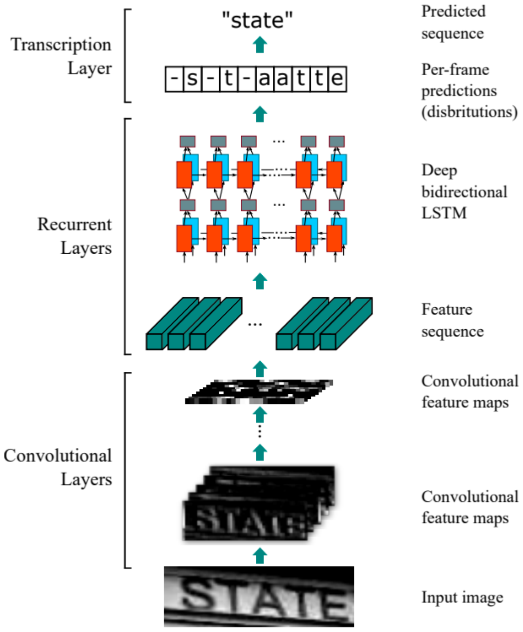


Figure 2: CRNN CTC Architecture integrating the key features of CNN, RNN, and CTC (Shi et al., 2016).

Shaw et al. (2008) used Hidden Markov Model (HMM) to recognize handwritten Devanagari words. It is based on segmentation-free approach or holistic approach, which extracts global features from an image thus reducing the overhead of segmentation. They also prepared a dataset of 39,700 handwritten words in Devanagari script. The correct classification rate, misclassification rate and rejection rate obtained on the test set were 80.2%, 16.3% and 3.5% respectively. Singh et al. (2011) proposed a Curvelet feature extractor with Support Vector Machine (SVM) and k-Nearest Neighbors (k-NN) classifiers based scheme for the recognition of handwritten Devanagari words. It was tested on a dataset of 28,500 handwritten words. Curvelet with k-NN gave overall better results than the SVM classifier and shown highest results (93.21%) accuracy on a Devanagari handwritten words set.

As Deep Learning techniques have developed, the OCR community has taken advantage of these developments to produce increasingly sophisticated and precise OCR models. While machine learning, the traditional method, yields high accuracy for OCR applications, new research indicates that deep learning techniques yield superior outcomes. Dutta et al. (2018) released a handwritten word dataset, called IIIT-HW-Dev, and benchmarked it using a CNN-RNN hybrid architecture. The proposed architecture consists of a spatial transformer layer (STN) followed by a set of residual convolutional blocks, which is proceeded by stacked bi-directional LSTM layers and ends with CTC layer for transcribing the labels. Dwivedi et al. (2020) have developed a Sanskrit specific OCR system to address complexities such as image degradation, lack of datasets and long-length words. They also introduced a dataset of 23848 annotated line images. The work has presented an attention-based LSTM model for reading Sanskrit characters in line images. It has a word error rate of 15.97% and a character error rate of 3.71%. Mondal and Jawahar (2022) used an attention-based encoder-decoder framework with a semantic module for an Indic handwritten text recognizer. The proposed framework achieved state-of-the-art results on handwritten texts of ten Indic languages.

While most works on Devanagari text recognition are primarily on Hindi documents, some efforts on Nepali handwritten text recognition can also be observed (Pant et al., 2012; Acharya et al., 2015; Pant and Bal, 2016; Pandey et al., 2017). Pant et al. (2012) prepared three datasets for Nepali Handwritten Characters, namely for numerals, vowels and consonants, and applied Multilayer Perceptron (MLP) and Radial Basis Function (RBF) classifiers. Recognition accuracy of 94.44% was obtained for numeral dataset, 86.04% for vowel dataset and 80.25% for consonant dataset. In all cases, RBF based recognition system outperformed MLP based recognition system but RBF based recognition system took little more time while training. Acharya et al. (2015) introduced a new publicly available image dataset for Devanagari script: Devanagari Handwritten Character Dataset (DHCD), consisting of 92 thousand images. They also proposed a deep learning architecture for recognition of those characters and obtained a test accuracy of 98.47%. Pant and Bal (2016) proposed a hybrid OCR system for printed Nepali text using the Random Forest (RF) Machine Learning technique. It incorporated two different approaches of OCR, the Holistic and the Character level recognition. The recognition rates of approximately 78.87% and 94.80% were achieved for character level recognition method and the Hybrid method respectively. Pandey et al. (2017) used Multi-layer Feed Forward Back Propagation Artificial Neural Network (ANN) for an OCR system for Nepali text in Devanagari script. Recognition accuracy of about 90% for simple words, 60% for complex words, and nearly 50% for handwritten words was achieved.

Among the notable research on the Nepalese Scripts are the works by O'Neill and Hill (2022), and Bati and Dawadi (2023). O'Neill and Hill (2022) introduced a model for Handwritten Text Recognition (HTR) of manuscripts written in Pracalit Script, trained on Transkribus with a PyLaia model based on ground truth generated from transcripts into Pracalit Unicode from four Nepalese manuscripts. Using 250 epochs, Transkribus trained a model with a CER on the training set of 2.6% and 0.1% on the validation set. Bati and Dawadi (2023) proposed a publicly available image database for the Ranjana Script, a script derived from the Nepal Script. They evaluated the Ranjana script Handwritten Character Dataset (RHCD) using Le-NET-5, AlexNET, ZFNET, and a proposed CNN model architecture. The proposed architecture achieved a testing accuracy of 99.73% for 64×64 pixel resolution at 53 epochs.

4. Methodology

The methodology employed in this work involved comprehensive data acquisition, preprocessing, dataset augmentation, model development, and evaluation. Following section explains these steps in detail.

4.1. Data Acquisition

To collect handwritten texts, forms were circulated among various individuals, organizations, and institutions, such as Nepal Lipi Guthi, Callijatra, etc. The sample collection forms, like the one shown in Figure 3, contain varying texts to be written. Images of 43 handwritten text samples were collected from 34 people who volunteered to fill up the form. The collected samples, along with additional samples extracted from handwritten and printed documents in the Nepal script, such as Pracalit Nepāl Lipiyā Varṇamālā (Nepal Lipi Guthi, 1992), were then manually segmented to produce 7,092 segments, each segment containing at most 3 words. A mapping of these segments to their corresponding transcriptions was carefully maintained, which comprised 3,302 unique words.

4.2. Preprocessing

The collected images further needed to be preprocessed to prepare a dataset for training the model. The preprocessing steps followed to normalize the text images are explained in the following sections.

RGB to Grayscale Conversion The collected images were RGB or RGBA as shown in Figure

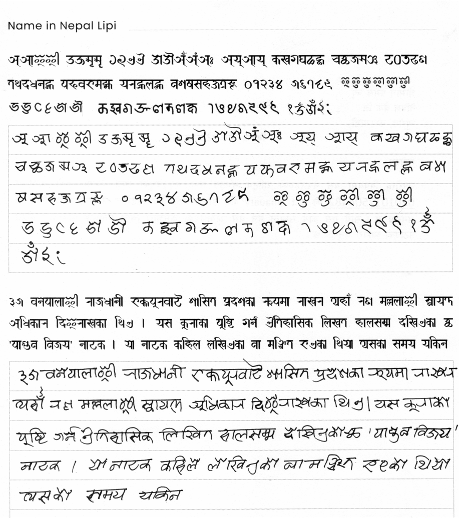

Name in Nepal Lipi

Figure 3: A Nepal script text sample collection form.

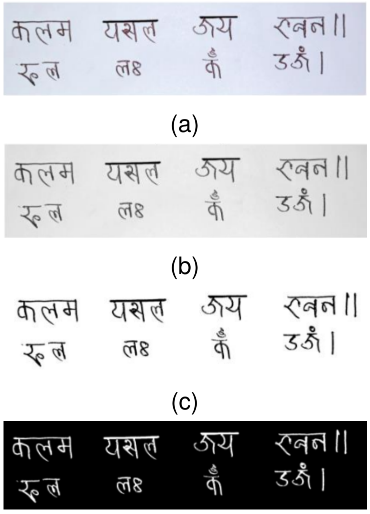
4a, and needed to be converted into a single channel grayscale image to discard all the color information. The conversion results in 2D images with distinct shadows and highlights of grays as shown in Figure 4b.

Normalization The grayscaled images were then normalized by transforming each pixel value to lie between 0 and 1.

Grayscale to Binary Conversion Normalized images were converted into binary images, which contain only two pixel values 0 representing black and 1 representing white, thereby separating the text from its background as shown in Figure 4c. This process is known as image binarization or thresholding. It works by finding a threshold value, T and making all the pixel values smaller than T as 0 and remaining pixel values greater than or equal to T as 1. We have used Adaptive and Otsu's Thresholding Techniques (Otsu, 1979), which automatically determine the optimal threshold value.

Inverse Binarization Next, the binarized images were inverted so that the text pixels are represented by 1s and the background pixels are represented by 0s as shown in Figure 4d. If B is a binarized image and IB is an inverted binarized image, then $IB(x, y) = 1 - B(x, y)$ where x, and y are the coordinates of a pixel in an image.

Noise Removal Noises in an image are the unnecessary pixels which may disturb the further processing like segmentation. Noises are removed by applying Median or Gaussian filters and morphological transformations. The noisy pixels are replaced by the median value and the the mean value of the neighbourhood pixels in a Median and a Gaussian filters respectively.



(a)

(b)

(c)

(d)

Figure 4: (a) RGB image, (b) Grayscale image, (c) Binary image, (d) Inverted binary image. Equivalent text in International Alphabet of Sanskrit Transliteration (IAST) format from left to right: kalama pasala jaya bhavana phala laḥ kaṁ ujaṃ; Translation: pen shop victory building fruit water tell permission.

4.3. Dataset Augmentation

As the Nepal Script is still under-resourced and not used by many, preparing a trainable dataset is challenging. We had to prepare the dataset ourselves with limited resources. However, the prepared dataset has a very limited amount of text samples, which cannot represent several conjuncts, modifiers, and handwriting styles properly, declining the overall performance of the model. Owing to this, we decided to augment the dataset, which in turn even helped us address different individualist styles and variations of handwritings. As the prepared dataset is not sufficient for the work, we performed 5-fold data augmentation to increase the dataset and improve the performance. After applying geometric transformations such as rotation, translation, scaling, and shearing, an augmented dataset with 35,460 samples was obtained. Using this technique can lower the chances of fitting a model too closely to the training data, leading to poor performance on new and unseen data. Moreover, it can help improve the model's ability to perform well on a variety of data, making it more generalizable, without simply memorising the idiosyncrasies of the dataset. The configurations used for this step are listed in Table 1.

Table 1: Data augmentation configurations.

Operation	Range (±)
Rotation	5°
Horizontal translation	4%
Vertical translation	4%
Shearing	15%
Scaling	10%

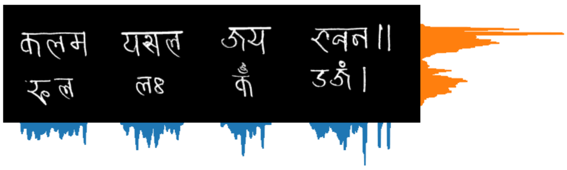


Figure 5: HPP and VPP of an inversed binary text image.

4.4. Line and Word Segmentation

As the work primarily involves a Nepal Script word recognizer, a text image needs to be segmented into lines and words. This step is primarily based on HPP and VPP. HPP is the sum of all column pixel values for each row and VPP is the sum of all row pixel values for each column as shown in Figure 5. Line and word segmentation was implemented with HPP and VPP respectively. Figure 6 represents line segmentation, while Figure 7 represents word segmentation.

4.5. Image Transformation and Character Encoding

The images were standardized to dimensions of 508×64 pixels (width×height). Padding was added to make the width uniform, and they were subsequently transformed to achieve dimensions of 64×508 pixels (width×height), aligning them with the timesteps of the RNN layers. Furthermore, a character set comprising 102 Nepal script Unicode characters (Unicode, Inc., 2023) along with special symbols was utilized.

4.6. Model Development

As discussed in section 2, our text recognizer model is inspired by the combination of Convolutional Recurrent Neural Network (CRNN) and Connectionist Temporal Classification (CTC). It accepts an inversed binarized image of dimension 64×508 pixels (width×height) along with its encoded transcription. The implementation consists of five CNN layers, three Bi-LSTM based RNN layers and a CTC layer as shown in Figure 8. The CNN Network extracts features of characters in the image which are fed into the RNN Network for learning the sequence and to give the character predictions at each time step. The transcription layer, which is based on CTC decodes the per time step predictions to calculate the loss to train the model and detect the text without the need for explicit character-level segmentation.

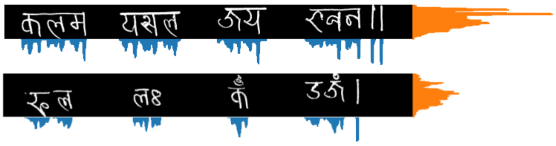


Figure 6: Segmented lines along with HPP and VPP.

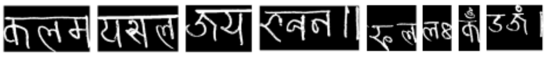


Figure 7: Segmented words from the input image.

5. Experimental Results

The augmented dataset, which contained 35,460 samples, was partitioned into training, validation and test sets with a split ratio of 70:15:15. The training set contained 24,822 samples, the validation set contained 5,319 samples and the test set contained 5,319 samples.

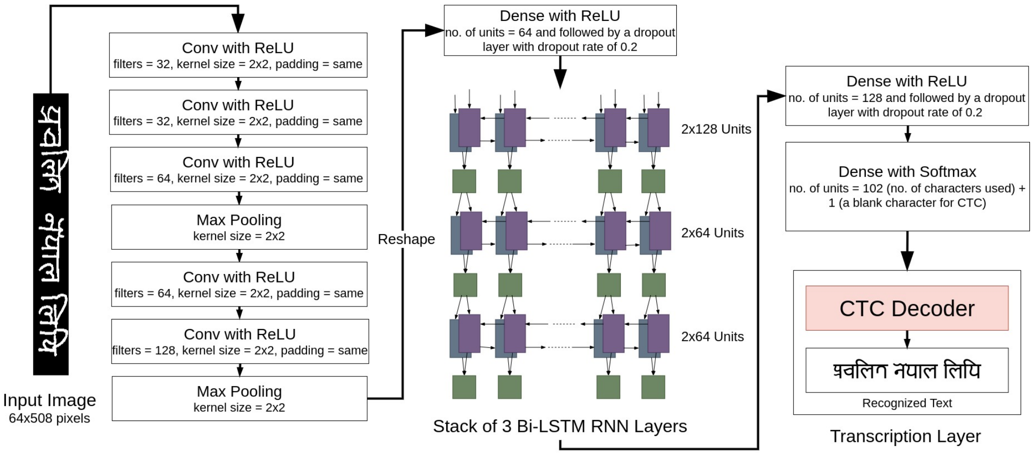
The model was trained for 100 epochs using the Adam optimizer with a learning rate of 0.001. The training was conducted on a Kaggle kernel utilizing a P100 GPU and took approximately 2.8 hours to complete. After training our model, it achieved Character Error Rate (CER) of 6.65% and Word Error Rate (WER) of 13.11%. Figure 9 shows Training and Validation CTC loss curve.

Due to the presence of variations, similar characters like 𑐖 (ja) and 𑑕 (5), 𑐟 (ta) and 𑑗 (7), modifier usage like 𑐎𑐾 (ke), 𑐎𑐿 (kai), 𑐎𑐸 (ku) and conjuncts like 𑐪 (mha), 𑐯 (lha) in the Nepal Script, achieving a high accuracy was challenging. Figure 10 shows the recognized results for a few sample images from our Nepal script text dataset. It also highlights errors caused by minor differences in characters.

Figure 11 shows the results of text recognition involving segmentation operations. Furthermore, the system recognized the computer font texts with only a few errors. However, it could not recognize some text due to incorrect line segmentation caused by inadequate line spaces.

6. Conclusion

In conclusion, this paper has provided a thorough process for developing an offline text recognition system for the Nepal Script using CRNN CTC



Conv with ReLU
filters = 32, kernel size = 2x2, padding = same
Conv with ReLU
filters = 32, kernel size = 2x2, padding = same
Conv with ReLU
filters = 64, kernel size = 2x2, padding = same
Max Pooling
kernel size = 2x2
Conv with ReLU
filters = 64, kernel size = 2x2, padding = same
Conv with ReLU
filters = 128, kernel size = 2x2, padding = same
Max Pooling
kernel size = 2x2
Input Image
64x508 pixels
Reshape
Dense with ReLU
no. of units = 64 and followed by a dropout layer with dropout rate of 0.2
2x128 Units
2x64 Units
2x64 Units
Stack of 3 Bi-LSTM RNN Layers
Dense with ReLU
no. of units = 128 and followed by a dropout layer with dropout rate of 0.2
Dense with Softmax
no. of units = 102 (no. of characters used) + 1 (a blank character for CTC)
CTC Decoder
प्रवलिग नयाल लिपि
Recognized Text
Transcription Layer

Figure 8: CRNN CTC based Nepal script text recognizer model architecture.

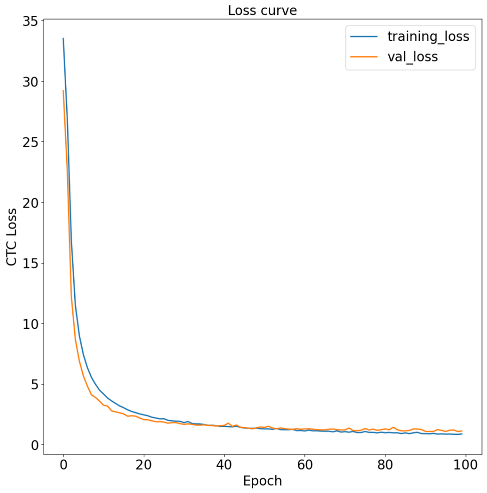


Figure 9: Training and validation CTC loss curve.

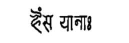
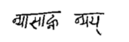
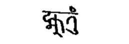
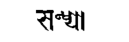
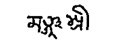
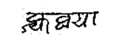
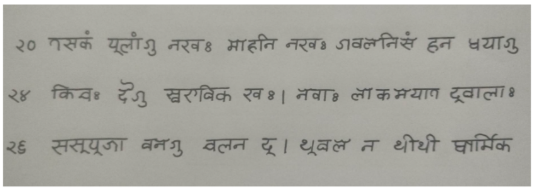
Image	Predicted Text	Ground Truth
	हृँस यानाः	ह्वँस यानाः
	ग्यासाङ्क ग्यय्	ग्यासाङ्क ग्यय्
	ह्मगुँ	ह्मगुँ
	सन्ध्या	सन्ध्या
	मञ्जुश्री	मञ्जुश्री
	ङ्गब्रया	ञ्जब्रया

Figure 10: Sample text predictions of our model with incorrect characters represented in red.

architecture. We prepared a dataset containing 7092 samples with 3,302 unique words. Various data augmentation techniques were applied to obtain an augmented dataset with 35,460 samples. Our model has achieved a CER of 6.65% and a WER of 13.11%. The dataset used for this work is available on Kaggle as Nepal Script Text Dataset (kaggle.com/ds/4763365) under CC BY-SA 4.0 license.



Predicted text

२० गसकं पूलांगु नखः माह्नि नखः गवलनिसं ह्नन धयागु २४ किवः द्ह ब्रगविक खः | नवाः लाकमयाग दालाः २६ ससूपूना वनगु वलन दू | थूवल न थीथी षार्मिक

Ground truth

२० गसकं पूलांगु नखः माह्नि नखः गवलनिसं ह्नन धयागु २४ किवः दँगु ब्रगविक खः | नवाः लाकमयाग दूवालाः २६ ससूपूजा वनगु वलन दू | थूवल न थीथी धार्मिक

Figure 11: The system recognized well-written handwritten texts, except for some similar characters and conjuncts.

Challenges associated with the Nepal Script text recognition include unavailability of proper datasets for the Nepal Script, difficulties in collecting samples for the dataset due to a limited number of people familiar with this script, the under-resourcing of the Nepal Script, lack of dedicated research in this field, and the complexities arising from the intricacies and complexities of the characters. Our system would be relevant for manuscripts, inscriptions, normal handwritten, and printed texts provided that the significant text samples are available to train the system. In the future, we aim to increase the dataset to include a wide range of text variations and explore segmentation-free text recognition. We believe that this work will serve as a stepping stone towards preserving and revitalizing the Nepal Script, ultimately helping in the preservation of Nepal Bhasa, an endangered and under-resourced language.

7. Acknowledgements

We are grateful to Nepal Lipi Guthi and Callijatra for their technical support and expertise. We would like to thank all the volunteers and participants who contributed their time and effort in providing Nepal Script handwritten text samples for preparing the dataset.

8. Bibliographical References

Shailesh Acharya, Ashok Kumar Pant, and Prashnna Kumar Gyawali. 2015. Deep learning based large scale handwritten devanagari character recognition. In *2015 9th International conference on software, knowledge, information management and applications (SKIMA)*, pages 1–6. IEEE.

Jen Bati and Pankaj Raj Dawadi. 2023. Ranjana script handwritten character recognition using cnn. *JOIV: International Journal on Informatics Visualization*, 7(3):984–990.

Kartik Dutta, Praveen Krishnan, Minesh Mathew, and CV Jawahar. 2018. Offline handwriting recognition on devanagari using a new benchmark dataset. In *2018 13th IAPR international workshop on document analysis systems (DAS)*, pages 25–30. IEEE.

Agam Dwivedi, Rohit Saluja, and Ravi Kiran Sarvadevabhatla. 2020. An ocr for classical indic documents containing arbitrarily long words. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 560–561.

Ajoy Mondal and CV Jawahar. 2022. Enhancing indic handwritten text recognition using global semantic information. In *International Conference on Frontiers in Handwriting Recognition*, pages 360–374. Springer.

Nepal Lipi Guthi. 1992. *Pracalit Nepāl Lipiyā Varṇamālā*. Nepal Lipi Guthi.

Nobuyuki Otsu. 1979. A threshold selection method from gray-level histograms. *IEEE transactions on systems, man, and cybernetics*, 9(1):62–66.

Alexander James O'Neill and Nathan Hill. 2022. Text recognition for nepalese manuscripts in pracalit script. *Journal of Open Humanities Data*, 8.

Ram Chandra Pandey, Babu Ram Dawadi, Suman Sharma, and Abinash Basnet. 2017. Dictionary based nepali word recognition using neural network. *Int. J. Sci. Eng. Res*, pages 473–479.

Ashok Kumar Pant, Sanjeeb Prasad Panday, and Shashidhar Ram Joshi. 2012. Off-line nepali handwritten character recognition using multilayer perceptron and radial basis function neural networks. In *2012 Third Asian Himalayas International Conference on Internet*, pages 1–5. IEEE.

Nirajan Pant and Bal Krishna Bal. 2016. Improving nepali ocr performance by using hybrid recognition approaches. In *2016 7th International Conference on Information, Intelligence, Systems & Applications (IISA)*, pages 1–6. IEEE.

Bikash Shaw, Ujjwal Bhattacharya, and Swapan K. Parui. 2014. Combination of features for efficient recognition of offline handwritten devanagari words. In *2014 14th International Conference on Frontiers in Handwriting Recognition*, pages 240–245.

Bikash Shaw, Swapan Kumar Parui, and Malayappan Shridhar. 2008. Offline handwritten devanagari word recognition: A holistic approach based on directional chain code feature and hmm. In *2008 International Conference on Information Technology*, pages 203–208. IEEE.

Baoguang Shi, Xiang Bai, and Cong Yao. 2016. An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. *IEEE transactions on pattern analysis and machine intelligence*, 39(11):2298–2304.

Connor Shorten and Taghi M Khoshgoftaar. 2019. A survey on image data augmentation for deep learning. *Journal of big data*, 6(1):1–48.

Brijmohan Singh, Ankush Mittal, MA Ansari, and Debashis Ghosh. 2011. Handwritten devanagari word recognition: a curvelet transform based approach. *International Journal on Computer Science and Engineering*, 3(4):1658–1665.

Kashinath Tamot. 1991. Nepālamā pracalit lipiko paricaya. *Madhuparka*.

Unicode, Inc. 2023. Newa Range: 11400-1147F. Accessed on: February 24, 2024.