**Library Progress International** *Print version ISSN 0970 1052*
*Vol.44, No.1s, January-June 2024: P.103-114* *Online version ISSN 2320 317X*
*DOI: https://doi.org/10.52710/lpi.44.1s.9*

*Original Article* *Available online at www.bpasjournals.com*

# An Efficient Transfer Learning Approach for Handwritten Historical Gurmukhi Character Recognition using VGG16: Gurmukhi_Hhdb1.0 Dataset

**[1]Harpal Singh*, [2] Simpel Rani, [3]Gurpreet Singh Lehal**

**Author's Affiliation:**
[1]Research Scholar, Punjabi University, Patiala 147002, India
E-mail: harpal.pup@gmail.com
[2]Professor, YCOE, Punjabi University Guru Kashi Campus, Talwandi Sabo, Punjab 151302, India
E-mail: simpel_jindal@rediffmail.com
[3]Senior Project Consultant, IIIT, Hyderabad 500032, Telangana
E-mail: gslehal@pbi.ac.in
***Corresponding Author: Reetu Verma**, Research Scholar, Punjabi University, Patiala 147002, India
E-mail: harpal.pup@gmail.com

**How to cite this article**: Singh H., Rani S. and Lehal G.S. (2024). An Efficient Transfer Learning Approach for Handwritten Historical Gurmukhi Character Recognition using VGG16: Gurmukhi_Hhdb1.0. *Library Progress International,* 44(1s), 103-114.

**ABSTRACT**
Historical manuscripts of Gurmukhi script hold immense historical and cultural significance. These manuscripts offer valuable insights into north Indian culture, politics, civilization and Sikhism. Preserving and analyzing these manuscripts is essential for gaining a deeper understanding of the past. Despite this, minimal attention was given to the recognition of characters in historical Gurmukhi manuscripts compared to modern text due to its unique challenges and scarcity of large and specialized datasets. To bridge this gap, we present benchmark dataset "Gurmukhi_HHdb1.0". This dataset comprises 87,181 images of 33 characters extracted from 6,340 pages of 43 historical Gurmukhi manuscripts. To recognize the characters of the Gurmukhi_HHdb1.0 dataset, authors developed an efficient transfer learning-based architecture by fine-tuning the VGG16 architecture. The entire dataset was split into 3 parts: 70% for training, 15% for validation and 15% for testing, with each subset containing approximately that percentage of the images. The authors achieved test accuracy of 99.77% and validation accuracy of 99.90% using the proposed approach. This showcases the efficacy of the proposed approach in accurately recognizing historical Gurmukhi characters.

**KEYWORDS**
Historical Gurmukhi manuscripts; Convolutional Neural Network; Dataset; VGG16; Character Recognition; Transfer Learning

## 1. Introduction

Ancient Gurmukhi manuscripts are instrumental in uncovering the history, literature, and cultural nuances of Punjab and its people. There are over seven thousand historical books with Gurmukhi content available worldwide. The average lifespan of a preserved manuscript is 300 to 400 years. Over the time, many artifacts accumulated on these manuscripts. To save these invaluable manuscripts, libraries such as the Punjab Digital Library have scanned numerous ancient manuscripts and uploaded them on web portals. However, the lack of machine-readable formats of these manuscripts hinders the extraction of information from these scans (Rani 2016; Memon et al. 2020). To address this issue and make these manuscripts searchable and accessible, character recognition is a crucial step. The recognition of the characters in Gurmukhi manuscripts presents numerous challenges, including complex character sets with intricate curves and loops, as well as the presence of noise, bleed through, and degraded paper and ink quality. Due to these factors, certain character pairs appear similar, such as ਟ and ਦ, ਤ and ੜ, ਬ and ਥ, ਤ and ਡ, ਦ and ਚ, ਟ and ੲ, ਯ and ਅ, ਰ and ਹ, and ਵ and ਞ. Some character pairs like ਸ and ਮ, ਪ and ਧ, and ਥ and ਖ exhibit similar physical structures without a headline. Additionally, variations in ink color, stains, font sizes and writing styles further compound the recognition task (Rani 2016).. To address these challenges effectively, a substantial dataset comprising diverse handwriting samples from a significant number of individuals is essential.

However, the progress in automated recognition and analysis of historical Gurmukhi manuscripts has been hindered by the scarcity of datasets specifically tailored for these unique handwritten characters. The significance of handwritten datasets in the advancing the OCR technology is elaborated in paper (Pal et al. 2012) as a means to understand cultural heritage using scripts. Many authors have developed the benchmark datasets for various Indian scripts, including Bangla (Bhattacharya and Chaudhuri 2005; Biswas et al. 2017; Das et al. 2015; Hasan et al. 2019; Rabby et al. 2019), Oriya (Bhattacharya and Chaudhuri 2005), Devanagari (Basu et al. 2007; Bhattacharya and Chaudhuri 2008; Das et al. 2012), Telugu (Das et al. 2012) and Kannada (Alaei et al. 2011). Additionally, HP Lab India has developed datasets for three Indic scripts (Agnihotri 2012; Agrawal 2004).. Regarding Gurmukhi script, one large and 7 small character datasets have been identified (Kaur and Rani 2017; Kumar et al. 2019). Notably, there is currently no dataset of handwritten characters from historical Gurmukhi manuscripts. To address this gap, we meticulously curated the "Gurmukhi_HHdb1.0" dataset, comprising 87,181 images encompassing 33 characters sourced from 6,340 pages of 43 historical Gurmukhi manuscripts.

Efforts to recognize handwritten characters in Gurmukhi script began late. Maximum attempts utilized handcrafted features and machine learning methods such as SVM (Aggarwal and Singh 2015; Kumar et al. 2017; Siddharth et al. 2011; Sinha et al. 2012; Singh et al. 2012), KNN (Kumar et al. 2014; Sinha et al. 2012), Random Forest (Kumar et al. 2018) and a Wavelet-based recognizer (Singh et al. 2012) for Gurmukhi character recognition. Very few attempts have been made using deep learning architectures (Kaur and Rani 2017; Kaur et al. 2022; Mahto et al. 2021) due to unavailability of dataset. Maximum efforts were dedicated to recognizing isolated characters in the Gurmukhi script, with minimal attention given to characters of historical Gurmukhi script due to its unique challenges. Historically, very few attempts (Rani and Lehal 2016;Kumar et al. 2018) have been made to recognize isolated characters from historical Gurmukhi script, relying on handcrafted features and machine learning approaches due to the unavailability of a large dataset. This paper represents our initial attempt to recognize isolated characters of Gurmukhi script using deep learning.

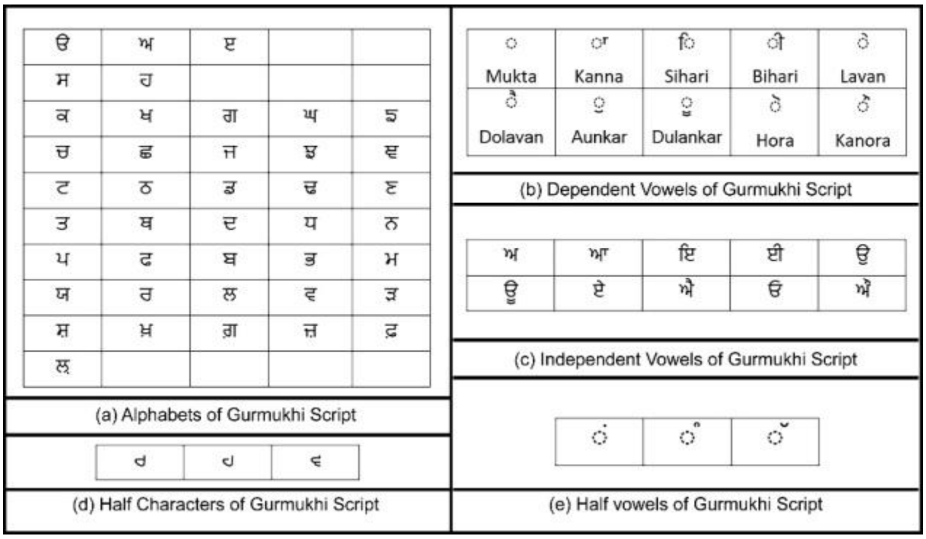
In this research paper, we conducted experiments using the Convolutional Neural Network (CNN) based VGG16 model. Initial testing on characters from the "Gurmukhi_HHdb1.0" dataset resulted in lower accuracy. To address this limitation and achieve higher accuracy, the authors proposed a fine-tuning approach for the existing CNN model. The proposed approach involved retraining the model on 61,146 images from the "Gurmukhi_HHdb1.0" dataset, with separate subsets of 13,010 and 13,025 images for validation and testing, respectively. The results demonstrate significant improvement, with our proposed aproach achieving 99.77% test accuracy.

The remaining paper is structured as follows: Sect. 2 introduces Gurmukhi script. Sect. 3 describes dataset development process. The proposed recognition scheme is discussed in Section 4. Design parameters and the experimental setup are presented in Sect. 5. Experimental results are described in Sect. 6.

Finally, conclusion is discussed in Sect. 7.

## 2. Gurmukhi Script

Gurmukhi script is one of official scripts of India and is primarily used for writing Punjabi. There are fifty characters in the Gurmukhi script, comprising thity eight consonants, nine vowels, three half vowels, three vowel carriers, and three half characters (Lehal and Singh 2001; Gurmukhi, Wikipedia). Figure 1 illustrates the forty-one alphabets present in the Gurmukhi script.


Mukta
Kanna
Sihari
Bihari
Lavan
Dolavan
Aunkar
Dulankar
Hora
Kanora

(a) Alphabets of Gurmukhi Script

(b) Dependent Vowels of Gurmukhi Script

(c) Independent Vowels of Gurmukhi Script

(d) Half Characters of Gurmukhi Script

(e) Half vowels of Gurmukhi Script

Figure 1. Alphabets of Gurmukhi Script

## 3. Dateset Development

This paper presents a dataset comprising isolated characters extracted from historical Gurmukhi manuscripts

### 3.1 Dataset Collection

The authors visited several villages, libraries, and Gurudwara sahibs to collect images for the dataset. The collected dataset comprises 6,340 pages that have been digitized from 43 historical Gurmukhi manuscripts. These manuscripts date from the 16th to 19th centuries. The documents were digitized using a Canon EOS 5D Mark IV digital camera and a scanner. Some documents were already available in digitized format. A detail of Gurmukhi_HHdb1.0 is presented in Table 1.

Table 1: Image Samples per Class in Gurmukhi_HHdb1.0

| Character | No. of Samples | Character | No. of Samples | Character | No. of Samples |
|---|---|---|---|---|---|
| ੳ | 2809 | ਜ | 3405 | ਨ | 2282 |
| ਅ | 3526 | ਝ | 1527 | ਪ | 3408 |
| ੲ | 2901 | ਟ | 3195 | ਫ | 2446 |
| ਸ | 3102 | ਠ | 914 | ਬ | 3520 |
| ਹ | 2035 | ਡ | 1313 | ਭ | 3102 |
| ਕ | 2551 | ਢ | 2035 | ਮ | 3481 |
| ਖ | 3354 | ਣ | 2224 | ਯ | 2551 |
| ਗ | 2556 | ਤ | 3697 | ਰ | 2349 |
| ਘ | 2626 | ਥ | 1213 | ਲ | 2154 |
| ਚ | 3627 | ਦ | 3525 | ਵ | 3160 |
| ਛ | 2488 | ਧ | 2420 | ੜ | 1685 |

The authors collected samples of 33 alphabets out of 35 alphabets of Gurmukhi script. The characters ਙand ਞare not used to write any word so their samples were not found in the scanned documents of Historical Gurmukhi manuscripts. Some images from Gurmukhi_HHdb1.0 are shown in Figure 2.

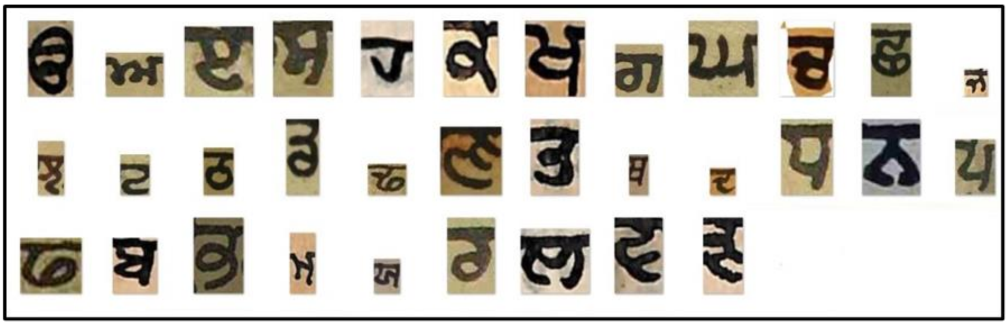


Figure 2. Image samples from Gurmukhi_HHdb1.0

### 3.2 Dataset Creation

To construct a character-level dataset, images of the alphabets were manually cropped from the 6,340 pages of 43 historical Gurmukhi manuscripts. In order to introduce variability in size and image quality, images were cropped using both computer systems and mobile phones. The Snipping and Shutter tools are used on Windows and Linux-based computer systems, respectively. Additionally, default cropping tools from various mobile phone manufacturers such as Samsung, Xiaomi, Sony, Lava, and Apple were utilized. These tools are operated on both IOS and Android systems to crop some of images, making the dataset more challenging and diverse.

### 3.3. Dataset Description

The "Gurmukhi_HHdb1.0" dataset comprises 87181 images representing 33 distinct alphabets. This dataset has been randomly splitted into 3 parts: the training, encompassing approximately 70% of the samples, the validation containing 15%, and the testing set, also comprising 15%. Division of samples in Gurmukhi_HHdb1.0 is given in Table 2.

Table 2: Distribution of samples in Gurmukhi_HHdb1.0 dataset

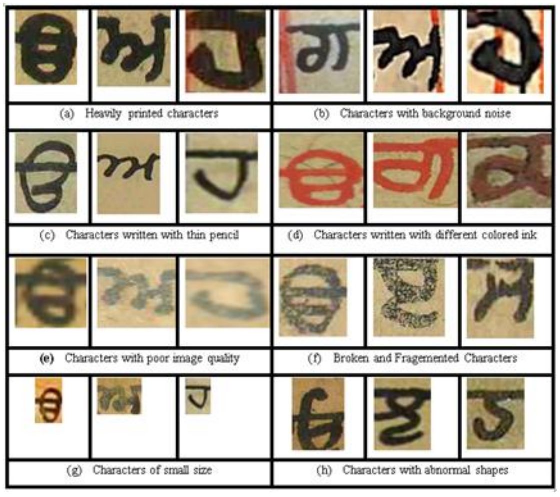
| No. of Classes | Samples in Training Dataset | Samples in Validation Dataset | Samples in Testing Dataset |
|---|---|---|---|
| 33 | 61146 | 13010 | 13025 |

### 3.4 Challenges in the Dataset

The authors cropped the dataset from 43 different manuscripts of Gurmukhi scripts, each of which presented unique challenges. The diversity in the collected dataset introduced various complexities in the process of character recognition. The characters were written with different types of pencils and ink, resulting in variations in line thickness, as illustrated in Figure 3(a) and Figure 1(c). Some samples in the dataset exhibited background noises such as stains, smudges, and stocks (Figure 3(b)). To make the dataset more challenging, samples written in multicolor ink were included, as shown in Figure 3(d). The dataset also includes images with poor quality, low resolution, blurriness, and other artifacts that may hinder accurate character analysis (Figure 1(e)). In addition, some samples contain broken and fragmented characters, as shown in Figure 3(f). Another challenge in the dataset was the presence of characters of different sizes; the size of some characters is large, while others are very small, as shown in Figure 3(g). Furthermore, some characters have abnormal shapes, adding an additional challenge to the character recognition process, as shown in Figure 3(h). The collected dataset presents diverse challenges, offering visual insight into variations in line thickness, background noises, broken characters, image quality, character shapes, and size disparities.

### 3.5 Data Augmentation

It is a technique used to produce synthetic samples from original dataset. In the proposed method, the Augmenter library of Python is employed for data augmentation (Bloice et al., 2023). Image transformations, such as shearing, random distortion, and rotation, were applied to the images in the test set, generating 3,000 augmented images per class. This augmentation process increased dataset's sample count to 186,181, which enable the model to handle images with random transformations. Furthermore, these operations prevent the model from overfitting and enhance its overall efficiency.



(a) Heavily printed characters
(b) Characters with background noise
(c) Characters written with thin pencil
(d) Characters written with different colored ink
(e) Characters with poor image quality
(f) Broken and Fragemented Characters
(g) Characters of small size
(h) Characters with abnormal shapes

Figure 3. Various Challenges of dataset

## 4. Proposed Recognition Scheme

CNN is a multilayer neural network designed to learn features directly from input image and minor preprocessing is required (Rahman et al. 2015. The proposed model employs transfer learning by fine-tuning a pretrained VGG16 network on our dataset to recognize the characters of historical Gurmukhi manuscripts. The VGG architecture achieved second position in ILSVRC2014 competition (Hinton 2012). VGG16 architecture has a simple and unified architecture with thirteen convolutions, five pooling layers and three fully connected layers as shown in Figure. 4.

Conv1, Conv2, and Conv3 consist of 64, 128, and 256 filters respectively, while both Conv4 and Conv5 have 512 filters, and each filter has a 3×3 kernel. The smaller size of the convolution filter and fewer receptive field channels reduce the computational costs (Balci et al. 2017). A 2×2 kernel is employed in all max-pooling layers. ReLU activation function (RAF) is applied to every convolutional layer and two dense layers, ensuring that only positive values are passed to the next stage of the network. The softmax activation function (SAF) is used at the output layer.

### 4.1. Proposed Architecture

Initially, the output layer of the VGG16 architecture was substituted with a new one to classify the characters of the historical Gurmukhi manuscript.

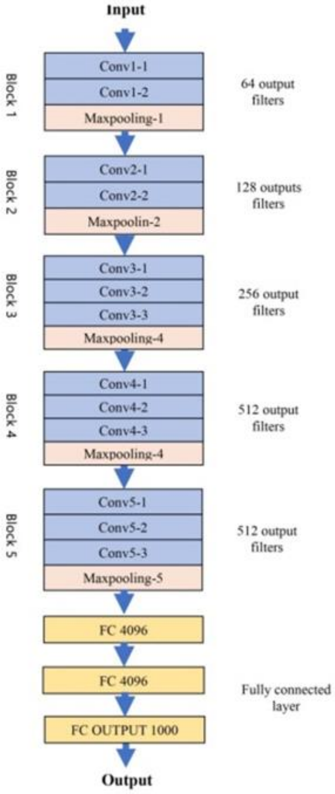


Figure 4. VGG16 architecture

The model was then retrained on our dataset, obtaining accuracies of 91.67% and 45.3% on training and testing set. To enhance the accuracy, VGG16 was fine-tuned by replacing its original dense layers with three new layers. Among these three dense layers, two consist of 512 and 256 units, respectively, with a RAF, while the last layer consists of 33 units with a SAF. The ReLU is a nonlinear activation function as given in Eq. (1).

$$\text{ReLU}(i) = \max(0), i \qquad (1)$$

It is preferred over the sigmoid and tanh due to its faster training with gradient descent (Nair and Hinton 2010) and its resistance to gradient vanishing problem. Due to simple mathematical operations of ReLU, it is computationally more efficient than Tanh and sigmoid.

Layers of first 3 blocks of the VGG16 network were set to be nontrainable to retain the pre-trained features and the remaining layers were fine-tuned.

To prevent overfitting in the network, L1 and L2 regularization methods with a regularization factor of 0.01 were employed. Additionally, image normalization was performed during the training process to enhance the stability and convergence of proposed architecture. The proposed architecture is illustrated in Figure 5.

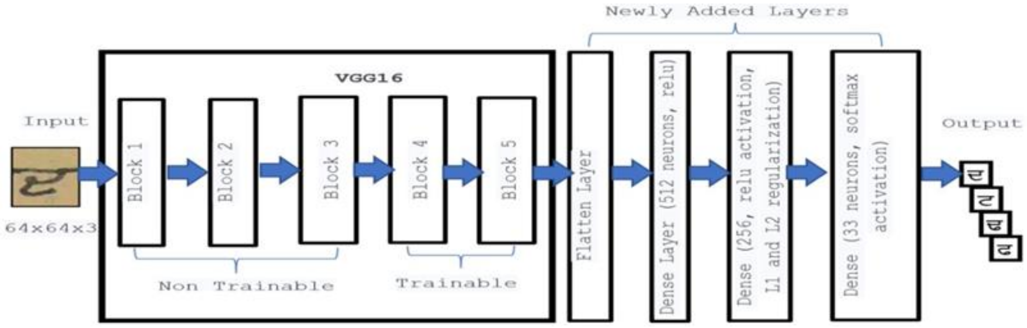

Newly Added Layers
VGG16
Input
64x64x3
Block 1
Block 2
Block 3
Block 4
Block 5
Non Trainable
Trainable
Flatten Layer
Dense Layer (512 neurons, relu)
Dense (256, relu activation, L1 and L2 regularization)
Dense (33 neurons, softmax activation)
Output

Figure 5: Proposed Architecture

## 5. Design Parameters and Experimental Setup

The proposed network is trained on 160,146 images, and validation and testing accuracy are evaluated on 13,010 and 13,025 images of the validation and testing sets of Gurmukhi_HHdb1.0, respectively. All images were resized to dimensions of 64×64×3. Image normalization was applied to all images. The architecture of the fine-tuned VGG16 model for recognizing Gurmukhi characters compromises several layers. The base VGG16 network, which has 14,714,688 parameters, outputs a shape of (None, 2, 2, 512). This is followed by a flatten layer, whuch convert the output to a shape of (None, 2048) with no additional parameters. Subsequently, a dense layer with 512 units and 1,049,088 parameters is added. Another dense layer with 256 units and 131,328 parameters follows. The final dense layer, with 33 units corresponding to the number of character classes, has 8,481 parameters.

The original VGG architecture, operating with an image size of 224×224×3, required 31 hours and 47 minutes to train on our dataset in Google Colab with an A100 GPU. To decrease the of trainable parameters and execution time, experiments were conducted with smaller image sizes, specifically 128×128×3, 64×64×3, and 32×32×3. Among these, better results were obtained with an image size of 64×64×3. The proposed model takes 8 hours and 10 minutes to train using this architecture in Google Colab with an A100 GPU. The proposed model has a total of 15,903,585 parameters, with 14,168,097 parameters being trainable and 1,735,488 parameters being non-trainable. Summary of hyper parameters is specified in Table 3.

Table 3: Hyper Parameters of Proposed Architecture

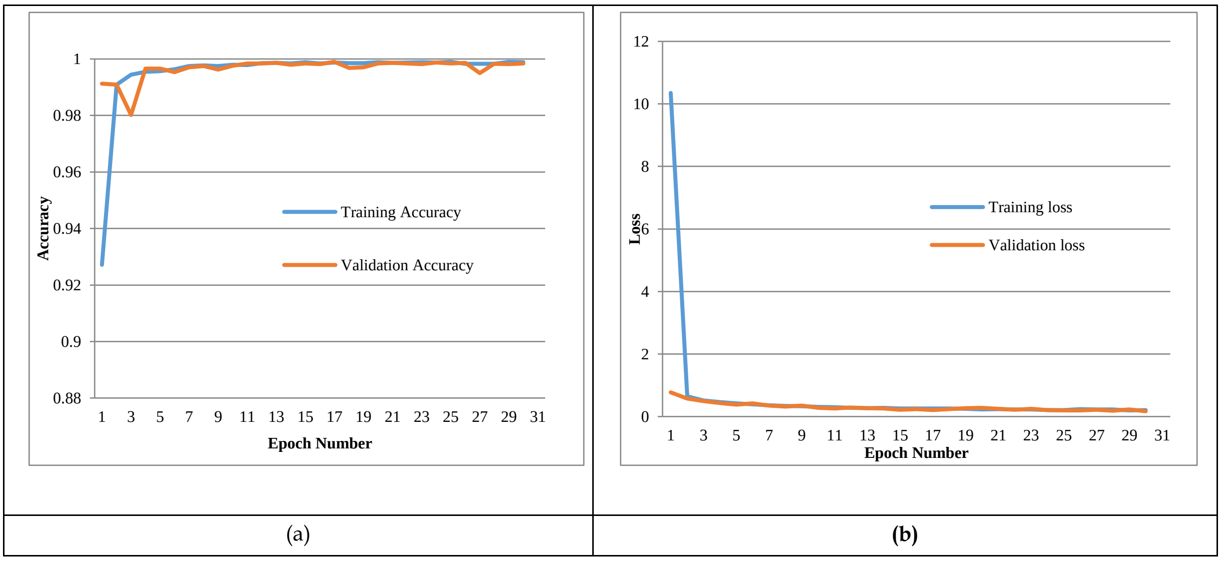
| | |
|---|---|
| **Optimizer** | SGD |
| **Learning Rate** | 0.001 |
| **Entropy:** | categorical cross entropy |
| **Training Batch sizes** | 32/64/128 |
| **Validation Batch sizes** | 32/64/128 |

## 6. Experimental Results

The model was trained on Google Colab using an A100 GPU for 30 epochs with batch sizes of 16, 32, and 64. Experiments were performed using RMSProp, Adam and SGD optimizers. The maximum training and validation accuracies obtained with SGD optimizer during training are summarized in Table 4. Figure 6 illustrates the model's training and validation accuracies across various epochs. The proposed architecture obtained testing accuracy of 99.77% with a batch size 32, thereby demonstrating its efficacy in recognizing characters from historical Gurmukhi manuscripts. The classification report of the proposed architecture with different batch sizes is given in Table 5. In addition, experiments with different optimizers were conducted and its effect on the classification report of the model is shown in Table 6.

*Table 4: Performance Metrics: Loss and Accuracy*

| Training Loss | Training Accuracy | Validation loss | Validation Accuracy |
|---|---|---|---|
| 0.1965473 | 0.9990023 | 0.169964448 | 0.999000 |



(a) **(b)**

Figure 6: Epoch Wise Training and Validation (a) Accuracy and (b) Loss

Table 5: Classification Report of Proposed Architecture with SGD Optimizer

| Batch Size | Accuracy | Metric | Precision | Recall | F1-Score | Support |
|---|---|---|---|---|---|---|
| 128 | 99.68% | MA | 99.63% | 99.55% | 99.59% | 13025 |
| | | WA | 99.68% | 99.68% | 99.68% | 13025 |
| 64 | 99.75% | MA | 99.65% | 99.77% | 99.71% | 13025 |
| | | WA | 99.76% | 99.75% | 99.76% | 13025 |
| 32 | **99.77%** | MA | **99.74%** | **99.78%** | **99.76%** | **13025** |
| | | WA | **99.77%** | **99.77%** | **99.77%** | **13025** |

Table 6 Classification Report of Proposed Architecture with Different Optimizer

| Optimizer | Accuracy | Metric | Precision | Recall | F1-Score |
|---|---|---|---|---|---|
| RMSProp | 98.07% | MA | 98.16% | 97.77% | 97.94% |
| | | WA | 98.13% | 98.07% | 98.08% |
| Adam | 98.87% | MA | 98.54% | 98.79% | 98.65% |
| | | WA | 98.90% | 98.87% | 98.88% |
| SGD | **99.77%** | MA | **99.74%** | **99.78%** | **99.76%** |
| | | WA | **99.77%** | **99.77%** | **99.77%** |

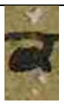
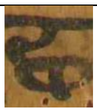
The proposed model misclassified only 30 images out of 13,025 images in the test dataset. Analysis of the confusion matrix, shown in Table 7, revealed that misclassifications occurred predominantly in cases where the shapes of both characters were similar or due to the presence of noise. The sample copies of misclassified characters and the analysis of misclassification is given in Table 8.

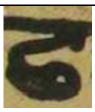
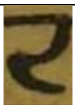
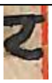
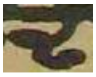
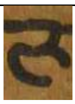
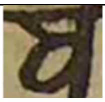
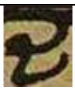
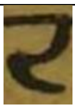
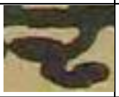
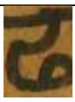
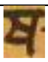
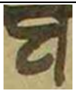
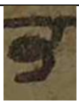
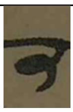
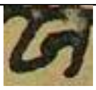
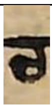
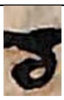
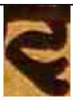
Table 7: Confusion Matrix

| Character | Total No. of Samples | No. of Samples correctly recognized | No. of Samples incorrectly recognized | Confused with Characters | | |
|---|---|---|---|---|---|---|
| ਉ | 421 | 421 | 0 | | | |
| ਅ | 532 | 532 | 0 | | | |

| | | | | | | |
|---|---|---|---|---|---|---|
| ੲ | 435 | 435 | 0 | | | |
| ਸ | 465 | 465 | 0 | | | |
| ਹ | 309 | 309 | 0 | | | |
| ਕ | 382 | 381 | 1 | ਚ(1) | | |
| ਖ | 503 | 503 | 0 | | | |
| ਗ | 383 | 383 | 0 | | | |
| ਘ | 393 | 393 | 0 | | | |
| ਚ | 544 | 544 | 0 | | | |
| ਛ | 373 | 371 | 2 | ਢ(1) | ਫ(1) | |
| ਜ | 510 | 510 | 0 | | | |
| ਝ | 193 | 193 | 0 | | | |
| ਟ | 479 | 477 | 2 | ਦ(1) | ਪ(1) | |
| ਠ | 137 | 137 | 0 | | | |
| ਡ | 198 | 198 | 0 | | | |
| ਢ | 309 | 307 | 2 | ਦ(2) | | |
| ਣ | 334 | 333 | 1 | ਨ(1) | | |
| ਤ | 555 | 554 | 1 | ਥ(1) | | |
| ਥ | 159 | 158 | 1 | ਬ(1) | | |
| ਦ | 529 | 521 | 8 | ਚ(1) | ਟ(6) | ਢ(1) |
| ਧ | 363 | 363 | 0 | | | |
| ਨ | 342 | 342 | 0 | | | |
| ਪ | 511 | 511 | 0 | | | |
| ਫ | 368 | 367 | 1 | ਢ(1) | | |
| ਬ | 528 | 524 | 4 | ਥ(3) | ੲ(1) | |
| ਭ | 465 | 463 | 2 | ਤ(1) | ਕ(1) | |
| ਮ | 522 | 522 | 0 | | | |
| ਯ | 382 | 381 | 1 | ਮ(1) | | |
| ਰ | 352 | 350 | 2 | ਚ(1) | ਡ(1) | |
| ਲ | 323 | 323 | 0 | | | |
| ਵ | 475 | 473 | 2 | ਟ(2) | | |
| ੜ | 251 | 251 | 0 | | | |
| **Total** | **13025** | **12095** | **30** | | | |

Table 8: Analysis of misclassified characters

| Actual Character | Predicted Character | Image | Reason of Misclassification |
|---|---|---|---|
| ਕ | ਚ |  | Due to the ink fill in the lower part, the model misclassified the character as "ਕ "instead of "ਚ" |
| ਛ | ਢ |  | Blurring in the lower part caused the circle to be misclassified as "ਢ "instead of "ਛ" |

| | | | |
|---|---|---|---|
| ਛ | ਫ |  | Overlapping lines in circular shapes merged the blank part with the top line, resulting in "ਫ "instead of "ਛ" |
| ਟ | ਦ |  | Clotting of ink created a circle bulge, leading to misclassification as "ਟ "instead of "ਦ" |
| ਟ | ਪ |  | A red page outline altered the character shape from "ਟ "to "ਪ" |
| ਢ | ਦ |  | Clots in the lower tip circle distorted the character shape, resulting in "ਦ "instead of "ਢ" |
| ਣ | ਨ |  | The written shape closely resembled predicted character “ਨ”, deviating from the actual character “ਣ” |
| ਥ | ਬ |  | Visually similar shapes of actual and predicted characters led to misclassification of ਬ instead of ਥ |
| ਦ | ਚ |  | A higher line in the lower right corner made the actual character "ਦ "visually similar to predicted "ਚ" |
| ਦ | ਟ |  | Ink covering the mid-right circle hid the blank space, which made the actual character "ਦ "resemble predicted "ਟ" |
| ਦ | ਢ |  | Bulging at the lower right end due to clotting made "ਦ "appear like "ਢ" |
| ਫ | ਢ |  | An extra bulge in the mid-left edge made the character "ਫ " resemble predicted "ਢ" |
| ਬ | ਥ |  | Similar shapes led to misclassification as "ਬ "instead of "ਥ" |
| ਬ | ਏ |  | The disconnected semicircle end caused misreading as "ਏ" instead of "ਬ." |
| ਭ | ਤ |  | Clot-filled circular shape led to misinterpretation as "ਭ "instead of "ਤ" |
| ਭ | ਕ |  | Incomplete bottom end caused reading as "ਕ "instead of "ਭ" |
| ਯ | ਮ |  | Blurred top line made the character "ਯ "appear like "ਮ" |
| ਰ | ਚ |  | Ink clot at the mid-left corner altered the shape from "ਰ "to "ਚ" |
| ਰ | ਡ |  | Large ink clot under the top link deformed the character "ਰ "to "ਡ" |
| ਵ | ਟ |  | Minimal space between lines led to misclassification as "ਟ " instead of "ਵ" |

## 7. Comparison with Existing Work

The proposed model's comparative analysis with existing methods (Rani 2016; Kumar et al. 2018), as illustrated in Table 9, establishes its superiority. Achieving a test accuracy of 99.71% on 87,181 images Gurmukhi_HHdb1.0, the proposed method significantly outperforms existing methods. In contrast, prior approaches obtained recognition accuracies of 96.21% (on 5600 images) and 95.91% (on 1140 images) using handcrafted features and machine learning-based classifiers. Training on a larger dataset of the proposed model demonstrated enhanced generalizability and efficiency in character recognition. The results showcase the efficacy of transitioning to deep learning methods for historical Gurmukhi manuscript character recognition.

Table 9: Comparison of proposed method with the existing methods for recognition of isolated characters of ancient Gurmukhi manuscript

| Authors | Dataset size | Feature extraction | Classification | Highest Accuracy |
|---|---|---|---|---|
| Rani, 2016 | 5600 | Discrete Cosine Transformations, Zoning, Gradient and Gabor filter | SVM (Linear Kernal) | 96.21% |
| Kumar et al., 2019 | 1140 | Zoning, Discrete Cosine Transformations and gradient features | k-NN, SVM, Decision Tree, Random Forest (adaptive boosting technique) | 95.91% |
| **Proposed Method** | **87,181** | **CNN** | **CNN** | **99.77%** |

## 8. Conclusion

In this study, authors presented a benchmark dataset, "Gurmukhi_HHdb1.0," consisting of isolated characters from historical Gurmukhi manuscripts. The dataset is curated from 6,340 pages across 43 different manuscripts and comprises 87,181 images of 33 distinct characters. This dataset is created for research purposes. Images are stored in raw .png format to enable researchers to conduct experiments from scratch. To recognize the characters of "Gurmukhi_HHdb1.0," authors developed an efficient transfer learning approach. They fine-tuned the VGG16 model by replacing its original dense layers with three new layers. The first layers of the first 3 blocks of the VGG16 network were set to be nontrainable to retain the pre-trained features, and the remaining layers were fine-tuned. In addition, L1 and L2 regularization methods with a regularization factor of 0.01 were employed to prevent overfitting in the network. To enhance the model's stability and convergence, we applied image normalization during the training process. The proposed model achieved 99.77% test accuracy and 99.90% validation accuracy on our dataset. The main contribution of this paper is the pioneer effort to introduce a benchmark dataset of isolated characters in historical Gurmukhi manuscripts and development of model for their recognition. The limitation of the proposed method lies in the manual selection and cropping of samples, which introduces potential bias. Additionally, the dataset is prepared from only 43 different manuscripts due to the unavailability of more manuscripts. Future work will extend the proposed method for compound character and recognition of complete line of historical Gurmukhi manuscripts.

### Conflict of Interest

All the authors declared that they have no conflict of interest in this work.

### Data Availability Statement

The authors generated their own dataset for the experimental study.

### Authors Contribution

***Harpal Singh***: Original Draft Writing; Experimental Work

***Simple Rani Jindal and Gurpreet Singh Lehal***: Supervision; Review; Final Draft

## References

[1] Aggarwal, A., & Singh, K. (2015, September). Handwritten Gurmukhi character recognition. In *2015 international conference on computer, communication and control (IC4)* (pp. 1-5). IEEE.

[2] Agnihotri, V. P. (2012). Offline handwritten Devanagari script recognition. *IJ Information Technology and Computer Science*, 8(1), 37-42.

[3] Agrawal, M., Bhaskarabhatla, A. S., &

Madhvanath, S. (2004, November). Data collection for handwriting corpus creation in Indic scripts. In *International Conference on Speech and Language Technology and Oriental COCOSDA (ICSLT-COCOSDA 2004), New Delhi, India (November 2004)*. Citeseer.

[4] Alaei, A., Nagabhushan, P., & Pal, U. (2011, September). A benchmark Kannada handwritten document dataset and its segmentation. In *2011 International Conference on Document Analysis and Recognition* (pp. 141-145). IEEE.

[5] Balci, B., Saadati, D., & Shiferaw, D. (2017). Handwritten text recognition using deep learning. *CS231n: Convolutional Neural Networks for Visual Recognition, Stanford University, Course Project Report, Spring*, 752-759.

[6] Basu, S., Chaudhuri, C., Kundu, M., Nasipuri, M., & Basu, D. K. (2007). Text line extraction from multi-skewed handwritten documents. *Pattern Recognition*, *40*(6), 1825-1839.

[7] Bhattacharya, U., & Chaudhuri, B. B. (2005, August). Databases for research on recognition of handwritten characters of Indian scripts. In *Eighth International Conference on Document Analysis and Recognition (ICDAR'05)* (pp. 789-793). IEEE.

[8] Bhattacharya, U., & Chaudhuri, B. B. (2008). Handwritten numeral databases of Indian scripts and multistage recognition of mixed numerals. *IEEE transactions on pattern analysis and machine intelligence*, *31*(3), 444-457.

[9] Biswas, M., Islam, R., Shom, G. K., Shopon, M., Mohammed, N., Momen, S., & Abedin, A. (2017). Banglalekha-isolated: A multi-purpose comprehensive dataset of handwritten bangla isolated characters. *Data in brief*, *12*, 103-107.

[10] Bloice, M. D. (2024). Augmentor. *Augmentor Documentation*. Retrieved May 20, 2024, from https://augmentor.readthedocs.io/en/master/userguide/mainfeatures.html

[11] Das, N., Basu, S., Sarkar, R., Kundu, M., & Nasipuri, M. (2015). An improved feature descriptor for recognition of handwritten bangla alphabet. *arXiv preprint arXiv:1501.05497*.

[12] Das, N., Reddy, J. M., Sarkar, R., Basu, S., Kundu, M., Nasipuri, M., & Basu, D. K. (2012). A statistical–topological feature combination for recognition of handwritten numerals. *Applied Soft Computing*, *12*(8), 2486-2495..

[13] Gurmukhi. (2024). In *Wikipedia*. Retrieved February 31, 2024, from https://en.wikipedia.org/wiki/Gurmukhi

[14] Hasan, M. M., Abir, M. M., Ibrahim, M., Sayem, M., & Abdullah, S. (2019, September). Aibangla: A benchmark dataset for isolated bangla handwritten basic and compound character recognition. In *2019 International Conference on Bangla Speech and Language Processing (ICBSLP)* (pp. 1-6). IEEE.

[15] Hinton, G. (2012). Coursera course - lecture 6e. Retrieved December 20, 2023, from http://www.cs.toronto.edu/tijmen/csc321/slides/lecture_slides_lec6.pdf

[16] Kaur, H., & Rani, S. (2017). Handwritten Gurumukhi character recognition using convolution neural network. *International Journal of Computational Intelligence Research*, 13, 933-943.

[17] Kaur, K., Chaudhuri, B. B., & Lehal, G. S. (2022, November). A Benchmark Gurmukhi Handwritten Character Dataset: Acquisition, Compilation, and Recognition. In *International Conference on Frontiers in Handwriting Recognition* (pp. 452-467). Cham: Springer International Publishing.

[18] Kumar, M., Jindal, M. K., & Sharma, R. K. (2017). Offline handwritten Gurmukhi character recognition: analytical study of different transformations. *Proceedings of the National Academy of Sciences, India Section A: Physical Sciences*, *87*, 137-143.

[19] Kumar, M., Jindal, M. K., Sharma, R. K., & Jindal, S. R. (2018). Offline handwritten numeral recognition using combination of different feature extraction techniques. *National Academy Science Letters*, *41*, 29-33.

[20] Kumar, M., Jindal, S. R., Jindal, M. K., & Lehal, G. S. (2019). Improved recognition results of medieval handwritten Gurmukhi manuscripts using boosting and bagging methodologies. *Neural Processing Letters*, *50*,

43-56.

[21] Kumar, M., Sharma, R. K., & Jindal, M. K. (2014). Efficient feature extraction techniques for offline handwritten Gurmukhi character recognition. *National Academy Science Letters*, *37*, 381-391.

[22] Kumar, M., Sharma, R. K., Jindal, M. K., Jindal, S. R., & Singh, H. (2019). Benchmark datasets for offline handwritten Gurmukhi script recognition. In *Document Analysis and Recognition: 4th Workshop, DAR 2018, Held in Conjunction with ICVGIP 2018, Hyderabad, India, December 18, 2018, Revised Selected Papers 4* (pp. 143-151). Springer Singapore..

[23] Lehal, G. S., & Singh, C. (2001). A technique for segmentation of Gurmukhi text. In *Computer Analysis of Images and Patterns: 9th International Conference, CAIP 2001 Warsaw, Poland, September 5–7, 2001 Proceedings 9* (pp. 191-200). Springer Berlin Heidelberg.

[24] Mahto, M. K., Bhatia, K., & Sharma, R. K. (2021). Deep learning based models for offline Gurmukhi handwritten character and numeral recognition. *ELCVIA Electronic Letters on Computer Vision and Image Analysis*, *20*(2), 69-82.

[25] Memon, J., Sami, M., Khan, R. A., & Uddin, M. (2020). Handwritten optical character recognition (OCR): A comprehensive systematic literature review (SLR). *IEEE access*, *8*, 142642-142668.

[26] Nair, V., & Hinton, G. E. (2010). Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)* (pp. 807-814).

[27] Pal, U., Jayadevan, R., & Sharma, N. (2012). Handwriting recognition in indian regional scripts: a survey of offline techniques. *ACM Transactions on Asian Language Information Processing (TALIP)*, *11*(1), 1-35.

[28] Rabby, A. S. A., Haque, S., Islam, M. S., Abujar, S., & Hossain, S. A. (2019). Ekush: A multipurpose and multitype comprehensive database for online off-line bangla handwritten characters. In *Recent Trends in Image Processing and Pattern Recognition: Second International Conference, RTIP2R 2018, Solapur, India, December 21–22, 2018, Revised Selected Papers, Part III 2* (pp. 149-158). Springer Singapore.

[29] Rahman, M. M., Akhand, M. A. H., Islam, S., Shill, P. C., & Rahman, M. H. (2015). Bangla handwritten character recognition using convolutional neural network. *International Journal of Image, Graphics and Signal Processing*, *7*(8), 42-49.

[30] Rani, S. (2016). *Recognition of Handwritten Gurmukhi Manuscripts.* Doctoral dissertation, Punjabi University Patiala, Punjab, India.

[31] Siddharth, K. S., Jangid, M., Dhir, R., & Rani, R. (2011). Handwritten Gurmukhi character recognition using statistical and background directional distribution. *Int. J. Comput. Sci. Eng.(IJCSE)*, *3*(06), 2332-2345..

[32] Singh, P., & Budhiraja, S. (2012). Offline handwritten Gurmukhi numeral recognition using wavelet transforms. *International Journal of Modern Education and Computer Science*, *4*(8), 34.

[33] Singh, S., Aggarwal, A., & Dhir, R. (2012). Use of Gabor Filters for recognition of Handwritten Gurmukhi character. *International Journal of Advanced Research in Computer Science and Software Engineering*, *2*(5).

[34] Sinha, G., Rani, R., & Dhir, R. (2012). Handwritten Gurmukhi character recognition using K-NN and SVM classifier. *International Journal of Advanced Research in Computer Science and Software Engineering*, 2(6), 288-293.