Images Classification of Gurumukhi Month's Name Images using Various Convolutional Neural Network Optimizers

Tajinder Pal Singh
Chitkara University
Sheifali Gupta
Chitkara University
Meenu Garg
Chitkara University
Deepika Koundal (koundal@gmail.com)
UPES: University of Petroleum and Energy Studies https://orcid.org/0000-0003-1688-8772
Atef Zaguia
Taif University

Research Article

Keywords: Gurumukhi script, Word text Images, Convolutional neural network, Optimizes, Performance Assessment

Posted Date: October 1st, 2021

DOI: https://doi.org/10.21203/rs.3.rs-921895/v1

Version of Record: A version of this preprint was published at Soft Computing on January 22nd, 2022. See the published version at https://doi.org/10.1007/s00500-022-06739-z.

Abstract

The Gurumukhi script has a complex structure for which text recognition based on an analytical approach can misinterpret the script. For error-free results in text recognition, the author has proposed a holistic approach based on classification of Gurumukhi month's name images. For this, a new convolutional neural model has been developed for automatic feature extraction from Gurumukhi text images. The proposed convolutional neural network is designed with five convolutional, three polling layers, one flatten layer and one dense layer. To validate the results of the proposed model, the dataset was self-created from 500 distinct writers. The performance of the model has been analyzed with 100 epochs, 40 batch sizes and different optimizers. The various optimizers that have been used for this experimentation are SGD, Adagrad, Adadelta, RMSprop, Adam, and Nadam. The experimental results show that the proposed CNN model performed best with Adam optimizer in terms of accuracy, computational time, F1 score, precision and recall.

1. Introduction

Nowadays, research on handwritten text image analysis and recognition for regional applications has become very popular as most of the essential information in our daily lives is going to be transferred electronically through computers. The electronically processed data is easy to handle and can be preserved for future use. In regional applications, handwritten text recognition is the most challenging task due to the complex structure of scripts like Gurumukhi. Usually, two different approaches, named analytical approach and holistic approach, are used for text recognition. An analytical approach to text recognition is basically a character level segmentation approach, where a complete handwritten text is segmented into small units called characters for recognition. On the other hand, the holistic approach is a segmentation-free approach, in which the contour or shape information of the text is transferred to an appropriate recognizer for recognition. In handwritten text with overlapped and touching characters, where it's hard to segment the characters, the holistic approach outperforms the analytical approach in terms of recognition accuracy. In the present work, the author proposed a holistic approach to handwritten Gurumukhi text recognition.

The recognition accuracy of handwritten text is also affected by the method used for feature extraction of text images. There are two methods of feature extraction. One is manual feature extraction and the other is automatic feature extraction. In manual feature extraction of a text image, features that are unique to the image are identified and the methods to extract them are implemented. Previously, various manual feature extraction methods were explored by researchers for text recognition. For example, (Sharma et al., 2008) and (Sharma et al., 2009) manually extracted the low level features (linearity, curliness, width, height, aspect ratio, slope, area, etc) and high level features (loops, crossings, straight lines, headlines, and dots) of online handwritten Gurumukhi characters in their proposed work and achieved 90.08% text recognition accuracy. (Dhir, 2010), has explored various moments like zernike moments, pseudo zernike moments, and orthogonal fourier-Melin moments for moment based invariant feature extraction of Roman and Gurumukhi characters. The best results were found using pseudo Zernike moments. (Kumar et al., 2011) compared the handwriting of various writers based on zoning, directional & diagonal features. (Kumar et al., 2014) proposed various feature extraction techniques like parabola curve fitting based features and power curve fitting based features for recognition of offline Gurumukhi characters and achieved 98.10% accuracy when tested on the kNN classifier. (Kumar et al., 2015) and (Verma & Sharma, 2017) both manually extracted 64 point features of online handwritten Gurumukhi characters in their research work for Gurumukhi text recognition (Kumar et al., 2017) used discrete wavelet transformations (DWT2), discrete cosine transformations (DCT2), fast fourier transformations, and fan beam transformations to obtain a feature set of offline handwritten Gurumukhi characters for text classification. (Singh et al., 2017) in their article, proposed a points feature, discrete fourier transformation features, and directional features for the recognition of the online handwritten Gurumukhi script. The proposed feature extraction method gives maximum character recognition accuracy of 97.1%.(Kumar & Gupta, 2017) achieved recognition accuracy of 99.3% on offline handwritten Gurumukhi characters using local binary pattern (LBP) features, directional features, and regional features. In this article, text classification is done using a deep neural network. (Mahto et al., 2018) recognized offline handwritten Gurumukhi characters based on histogram oriented gradient (HOG) and pyramid histogram oriented gradient (PHOG) features. (Kumar et al., 2018) and (Sakshi et al., 2018) in their research work, prove that the writer identification in Gurumukhi text can be performed with an accuracy of 89.85% and 81.75% respectively, when a combination of various features extraction methods like zoning, transitions, and peak extent based features, centroid, parabola curve fitting, and power curve fitting methods are used. (Kumar et al., 2019) recognized works written by distinct writers in the 18th to 20th century using zoning, discrete cosine transformations, and gradient feature extraction methods. Based on these features, the authors have achieved recognition accuracy of 95.91%. (Garg et al., 2019) and proposed zoning, diagonal, peak extent based features (horizontally and vertically) and shadow features with a combination of k-NN, decision tree, and random forest classifier for recognition of degraded handwritten Gurumukhi characters. In their article, the authors achieved a maximum recognition accuracy of 96.03% using a random forest classifier with zoning and shadow features. (Kumar et al., 2020) worked on a dataset of offline handwritten Gurumukhi characters and numerals to evaluate the performance of various classifiers based on peak extent features, diagonal features, and centroid features.

Manual feature extraction can be replaced by a deep neural network for automatic feature extraction. The authors used deep neural networks for text classification. For example, (Jindal et al., 2020) proposed a technique that used a deep convolutional neural network to analyze 3500 Gurumukhi characters for feature extraction. The network obtained an accuracy of 98.32 percent on the training set and 74.66 percent on the test set using two convolutional and two pooling layers.

With the passage of time, the research on deep convolutional neural networks has progressed by introducing numerous ways to achieve astonishing results. This research is based on the choice of an optimal hyper-parameter to reduce the error rate of the network. The choice of optimal weight parameters to reduce network loss is a critical issue for which researchers have introduced various deep learning optimizers. These optimizers show a considerable enhancement in the performance of deep neural networks. In the context of Gurumukhi month's name recognition, the performance assessment of the optimizers on convolutional neural networks has not been done previously. In the present work, six deep learning optimizers, SGD, Adagrad, Adadelta, RMSprop, Adam, and Nadam are deployed to train the proposed convolutional neural network. For comprehensive performance assessment of the proposed model using various optimizers, results of validation accuracy/loss, computational time, F1 score, precision and recall have been compared.

In brief, the following are the major contributions of this research work:

1. The dataset has been prepared for 24 different classes of Gurumukhi months written by 500 different writers.
2. The author has developed a new convolutional neural network for classification of Gurumukhi months on a designed dataset.
3. A comparative performance analysis has been done using various optimizers and training batch sizes.

The remaining part of this article is divided into two different sections. Section 2 is on materials and methods, including experimental datasets, the proposed CNN architecture, deep learning optimizers, and experimental setup. In Sect. 3 of this article, obtained results from the experiment based on various parameters like accuracy, computational time, F1 score, precision, and recall are presented.

2. Material And Methods

This section contains the detailed information about the dataset preparation, proposed CNN model architecture and specifications of different optimizers used for simulation.

2.1 Dataset Preparation

The author used a self-prepared dataset of handwritten Gurumukhi month’s name to demonstrate the performance of proposed convolutional neural networks with various optimizers. The dataset has been prepared for 24 different classes of Gurumukhi months written by 500 different writers. The 500 different writers selected for dataset preparation belong to different age groups, genders and professions. The sample sheets of prepared datasets from two different writers are shown in Fig. 1.

On the sample sheet, each writer wrote twice a single month's name in the different blocks drawn on the sheets. Hence, for 24 different classes of Gurumukhi months, each writer wrote 48 handwritten words as shown in Fig. 1. For the entire dataset, 1000 words or samples for each class in 24 different classes of Gurumukhi months were collected, which resulted in 24000 handwritten words on paper documents.

These paper documents are further converted into digital form using an OPPO F1s smart phone. After digitization, an image containing the names of all the Gurumukhi months was cropped into 48 text images of size 50x50 using the MATLAB cropping tool as shown in Fig. 2. The same operation was applied to all digitized images, which resulted in 24000 text images in the dataset.

To train our proposed model, randomly, 80% of the labeled samples in the 24000 text images dataset were used and the remaining 20% of labeled samples were used for model testing. The details of the number of randomly selected dataset samples from each class of Gurumukhi month’s dataset for training and testing of the proposed model are shown in Table 1.

Table 1
Training and Testing Samples for Each Class of Gurumukhi Months Dataset

Sr No.	Class Name in English	Class Name in Gurumukhi	Number of Training Samples	Number of Testing Samples
1.	Vaisakh	□□□□□	795	205
2.	Jeth	□□□	801	199
3.	Harh	□□□□□	799	201
4.	Sawan	□□□□	782	218
5.	Bhado	□□□□□	802	198
6.	Assu	□□□□	809	191
7.	Katak	□□□□	813	187
8.	Magar	□□□□	789	211
9.	Poh	□□□	798	202
10.	Magh	□□□	788	212
11.	Phagun	□□□□	802	198
12.	Chet	□□□	819	181
13.	January	□□□□□	770	230
14.	February	□□□□□	817	183
15.	March	□□□□	804	196
16.	April	□□□□□□	786	214
17.	May	□□	803	197
18.	June	□□□	822	178
19.	July	□□□□□	797	203
20.	August	□□□□	788	212
21.	September	□□□□□	805	195
22.	October	□□□□□□	804	196
23.	November	□□□□□	808	192
24.	December	□□□□□	799	201
Total			19200	4800

2.2 Proposed CNN Model Architecture

The architecture of the proposed CNN model with complete specification of its convolutional layers and pooling layers is presented in Fig. 3.

As per the figure, the proposed CNN model has 5 convolutional layers and 3 pooling layers. For the proposed CNN model, all convolutional layers are designed for a filter size of 3x3 with variable numbers of filters in each layer. For example, the numbers of filters in the first convolutional layer are 32, for the second and third convolutional layer are 64, and for the fourth and fifth convolutional layer are 128. In the pooling layers of the model, the first pooling layer has a filter size of 3x3 and for the second and third pooling layers it is 2x2. The features from all layers are passed to the fully connected layer that employs softmax as the activation function and gives the final output of the network. The architectural details of the proposed CNN are given in Table 2.

Table 2
Architectural details of the proposed CNN

No of layers	Convolutional layer (ReLu Activation function)			Pooling layer			Flatten		Dense			
	Input Feature Map	Output Feature Map	Training Parameter	Input Feature Map	Output Feature Map	Dropout	Input Feature Map	Output Feature Map	Input Feature Map	Output Feature Map	Activation function	Training Parameter
L_1	50*50*1	50*50*32	320	50*50*32	16*16*32	0.25	4*4*128	2048	2048	1024	ReLU	2098176
L_2	16*16*32	16*16*64	18496	16*16*64	8*8*64	0.25			1024	24	Softmax	24600
L_3	16*16*64	16*16*64	36928	8*8*128	4*4*128	0.25						
L_4	8*8*64	8*8*128	73856	——	——	——						
L_5	8*8*128	8*8*128	147584	——	——	——						

2.3 Specifications of Different Optimizers Used for Simulation

For classification of Gurumukhi month's name images, the proposed model is simulated using various deep learning optimizers. The names of optimizers are SGD, Adagrad, Adadelta, RMSprop, Adam, and Nadam. The learning specifications of the deep learning optimizers are given below in Table 3.

Table 3
Deep learning Optimizers' Learning Specifications

Optimizers	Specifications
SGD	learning rate = 1.E-03,, momentum = 0.9, nesterov = False, decay = learning rate / EPOCHS
Adagrad	learning rate = 1.E-03, epsilon = 1e-07, decay = learning rate / EPOCHS
Adadelta	learning rate = 1.E-03, rho = 0.9, epsilon = 1e-07, decay = learning rate / EPOCHS
RMSprop	learning rate = 1.E-03, rho = 0.9, epsilon = 1e-07, decay = learning rate / EPOCHS
Adam	learning rate = 1.E-03, beta1 = 0.9,beta2 = 0.999, epsilon = 1e-07, decay = learning rate / EPOCHS
Nadam	learning rate = 1.E-03, beta1 = 0.9,beta2 = 0.999, epsilon = 1e-07, decay = learning rate / EPOCHS

3. Results Analysis For Different Optimizers

For training the proposed model, back propagation was performed using two different batch sizes of 20 and 40. In view of better model performance, other parameters like learning rate, dropout and epochs were fixed at 1e-3, 0.25 and 100 respectively. The experiment using the proposed model was performed on the Tesla K80 GPU on Google Colab with its limited use of 12 maximum hours of continuous training. For each epoch, the training time also varied, depending upon the traffic on the GPU at that time.

This section describes the results of the experiment performed on Gurumukhi month's name dataset using a proposed CNN model with six different optimizers.

3.1 Analysis of Computational Time

For evaluating the performance of the proposed model, the authors computed the training time for each optimizer. Table 4, given below shows the total time required for training the proposed model on different optimizers.

Table 4
Computational Time Comparison for Each Optimizer

	SGD (sec)	Adagrad (sec)	Adadelta (sec)	RMSprop (sec)	Adam (sec)	Nadam (sec)
Epoch 1	56	27	27	28	27	29
Epoch 2	9	10	10	9	9	10
—	—	—	—	—	—	—
—	—	—	—	—	—	—
Epoch 100	9	10	9	10	9	9
∑Total training time in (min)	16.5167	15.8333	15.65	15.5	15.7833	16.96667

From Table 4, it can be analyzed that the least training time is required by the proposed model using the RMSprop, Adadelta and Adam optimizers for classification of Gurumukhi month's name images. On the other hand, using the Nadam optimizer on the proposed model, the training time required is more as compared to other optimizers. It has also been noted that there is a minimum training time difference between the different optimizers used in this experimentation. With that, the optimizer's performance with respect to its computation time is also affected by the traffic on Google Colab while training the proposed model with GPU. Hence, while choosing the optimizer for the text image classification, computation time can't be a major factor.

3.2 Analysis of Class wise Accuracy

The image classification accuracy results of Gurumukhi month's name dataset on the proposed model using six different optimizers are presented in Table 5. In the given table, the results are shown for each month's class accuracy of the proposed model on 100 epochs with batch size 40.

Table 5
Class wise Accuracy (in %) for Different Optimizers

	SGD	Adagrad	Adadelta	RMSprop	Adam	Nadam
April	**100.00**	99.15	95.48	**100.00**	**100.00**	99.98
Assu	99.98	99.63	95.77	99.98	**100.00**	99.98
August	99.96	98.42	94.73	99.98	99.96	99.96
Bhado	99.98	98.94	95.67	99.98	99.98	99.96
Chet	99.96	98.02	96.29	99.92	99.98	99.92
December	99.96	98.27	95.17	**100.00**	99.98	99.98
February	99.94	98.94	91.88	99.98	99.98	99.96
Harh	**100.00**	99.69	90.75	99.96	99.98	99.98
Januray	99.96	98.52	94.04	99.96	99.98	99.94
Jeth	99.92	98.60	94.54	99.94	99.96	99.92
July	99.96	98.85	93.35	99.94	99.98	99.96
June	**100.00**	99.79	88.60	99.96	**100.00**	99.94
Katak	**100.00**	98.63	94.21	**100.00**	99.98	99.98
Magar	**100.00**	99.13	95.21	**100.00**	**100.00**	99.96
Magh	99.96	98.77	89.98	99.96	99.96	99.94
March	99.98	98.44	91.35	**100.00**	**100.00**	99.96
May	99.73	96.81	93.96	99.96	99.92	99.96
November	99.98	99.75	96.67	99.98	**100.00**	**100.00**
October	99.96	98.42	94.92	99.98	99.98	99.96
Phagun	99.90	98.25	86.90	99.98	**100.00**	99.94
Poh	99.98	99.23	95.29	**100.00**	**100.00**	**100.00**
Sawan	99.75	97.85	94.71	99.96	99.92	99.94
September	99.98	99.54	96.10	**100.00**	99.98	99.98
Vaisakh	99.94	97.00	94.06	99.90	99.96	99.94

The results shows that, SDG optimizer gives its best accuracy in 100% on five different classes April, Harh, June, Katak and Magar of Gurumukhi months dataset. On the other hand, when RMSprop simulated Gurumukhi months dataset using proposed CNN model, it has given best accuracy in 100% on seven different classes April ,December, Katak, Magar ,March, Poh and September of Gurumukhi months dataset.

The results depicted that, Adam optimizers outperformed other optimizers by giving best accuracy of 100% on eight different classes April, Assu, June, Magar, March, November, Phagun and Poh of Gurumukhi months dataset. Where Nadam only gives best accuracy on only two classes November Poh of Gurumukhi months dataset

3.3 Analysis of Overall Accuracy

Further in this section, the training and validation accuracy/loss graphs of the proposed model on various optimizers are shown in Fig. 4.

Here, Fig. 4 (a) presents the training and validation accuracy /loss graphs for SGD optimizer. The graph shows that the maximum validation accuracy of 99.37% has been achieved, when the proposed CNN model is simulated using SGD optimizer.

Figure 4 (b) presents the training and validation accuracy /loss graphs for Adagard optimizer. The graph depicts the maximum validation accuracy of 84.31%, when the proposed CNN model is simulated using Adagard SGD optimizer.

The presents the training and validation accuracy / loss graphs for Adadelta optimizers is shown in Fig. 4 (c). According to the figure, the validation accuracy that Adadelta optimizers have achieved on the proposed model is around 24.81%, which is the minimum validation accuracy obtained by Adadelta optimizers on the CNN model as compared to other optimizers.

Figure 4(d) shows the presents the training and validation accuracy /loss graphs for RMSprop optimizers. The graph shows that the validation accuracy using RMSprop on the CNN model has been achieved at around 99.65%. This is the second maximum validation accuracy achieved by RMSprop optimizers when tested using the proposed CNN model on a text image dataset.

An Adam optimizer has achieved maximum validation accuracy as compared to other optimizers, when tested on a text image dataset using the proposed CNN model as shown in Fig. 4 (e). The value of the obtained validation accuracy is 99.73%.

The presents the training and validation accuracy/loss graphs for the Nadam optimizer is presented in Fig. 4 (f). As per the figure, the Nadam optimizer has obtained a validation accuracy of around 99.50% on the proposed CNN model.

4. Results Analysis For Different Batch Sizes

Batch size is one of the important factors which can affect the learning of a CNN model. To test the performance of the proposed model, an experiment was performed using different batch sizes of 20 and 40 in the present work.

4.1 Analysis of Overall Accuracy and Loss

Table 6 given below, shows the comparative analysis in terms of training/validation accuracy and training/validation loss of the proposed model on different optimizers and batch sizes.

Table 6
Proposed CNN Model's Comparative Analysis with Different Batch Sizes

	Batch Size = 20				Batch Size = 40			
Parameters Optimizer	**Training Accuracy**	**Validation Accuracy**	**Training Loss**	**Validation Loss**	**Training Accuracy**	**Validation Accuracy**	**Training Loss**	**Validation Loss**
SGD	0.977	0.9973	0.0734	0.0183	0.9744	0.9937	0.0751	0.0267
Adagrad	0.5975	0.839	1.2471	0.4892	0.6462	0.8431	1.1001	0.4632
Adadelta	0.1386	0.2481	3.5092	2.4985	0.1391	0.2481	3.523	2.5076
RMSprop	0.9804	0.9954	0.0711	0.023	0.9886	0.9965	0.042	0.0193
Adam	0.9878	0.9973	0.0432	0.0225	**0.99**	**0.9973**	**0.0313**	**0.0188**
Nadam	0.9836	0.9956	0.0536	0.0266	0.9887	0.995	0.0358	0.0302

From Table 6, it's clear that all models with batch size 40 give better and more stable testing performance, that's why batch size 40 is chosen in the present work.

5. Analysis Of Different Confusion Matrix Parameters At Batch Size 40

In this section, performance assessment of the various optimizers has been done on the proposed CNN model in classification of Gurumukhi month's name images. Confusion matrix parameters generated while training the proposed CNN model is considered for the performance assessment of the optimizers. The various performance parameters like F1 score, precision and recall are used for this analysis.

5.1 F1 Score Results Analysis

Table 7 given below presented the F1 score results analysis, for performance assessment of the various optimizers on the proposed CNN model.

Table 7
F1 score for Each Class for different Optimizers

	SGD	Adagrad	Adadelta	RMSprop	Adam	Nadam
April	1.0000	0.9044	0.1933	1.0000	1.0000	0.9977
Assu	0.9974	0.9516	0.1714	0.9974	1.0000	0.9974
August	0.9953	0.8468	0.3659	0.9976	0.9953	0.9953
Bhado	0.9975	0.8547	0.0370	0.9975	0.9975	0.9949
Chet	0.9944	0.6494	0.4067	0.9889	0.9972	0.9890
December	0.9950	0.8168	0.2658	1.0000	0.9975	0.9975
February	0.9918	0.8468	0.3011	0.9973	0.9973	0.9945
Harh	1.0000	0.9631	0.2975	0.9950	0.9975	0.9975
Januray	0.9957	0.8583	0.2704	0.9957	0.9978	0.9935
Jeth	0.9899	0.8154	0.2762	0.9924	0.9949	0.9899
July	0.9950	0.8780	0.2958	0.9926	0.9975	0.9951
June	1.0000	0.9722	0.2831	0.9944	1.0000	0.9916
Katak	1.0000	0.8235	0.2147	1.0000	0.9973	0.9973
Magar	1.0000	0.8950	0.1786	1.0000	1.0000	0.9952
Magh	0.9953	0.8656	0.2023	0.9953	0.9953	0.9929
March	0.9975	0.8101	0.2095	1.0000	1.0000	0.9949
May	0.9681	0.6845	0.2487	0.9949	0.9899	0.9949
November	0.9974	0.9679	0.3939	0.9974	1.0000	1.0000
October	0.9949	0.7711	0.1921	0.9975	0.9974	0.9949
Phagun	0.9875	0.7941	0.2556	0.9975	1.0000	0.9924
Poh	0.9975	0.9063	0.2153	1.0000	1.0000	1.0000
Sawan	0.9718	0.7253	0.1533	0.9954	0.9908	0.9932
September	0.9974	0.9447	0.1221	1.0000	0.9974	0.9974
Vaisakh	0.9927	0.6712	0.1493	0.9879	0.9951	0.9927
Overall F1	0.9938	0.8424	0.2374	0.9964	**0.9973**	0.9950

From Table 7, it is clear that the proposed CNN model has given the best overall F1 score on the text image dataset, when tested using the Adam optimizer. The value of the overall F1 score using the Adam optimizer is around 0.9973, which is the highest of the overall F1 scores of other optimizers.

The value of the worst overall F1 score is 0.2374, which is using the Adadelta optimizer as per Table 7.

5.2 Precision Results Analysis

For the precision results analysis of different optimizers on the proposed CNN model, Table 8 is presented below.

Table 8
Precision Results for Each Class for Different Optimizers

	SGD	Adagrad	Adadelta	RMSprop	Adam	Nadam
April	1.0000	0.9023	0.4727	1.0000	1.0000	0.9953
Assu	0.9948	0.9779	0.3889	0.9948	1.0000	1.0000
August	1.0000	0.7394	0.3904	0.9953	1.0000	0.9953
Bhado	0.9950	0.9804	0.2222	0.9950	0.9950	1.0000
Chet	1.0000	0.9778	0.5126	0.9944	1.0000	0.9890
December	1.0000	0.7341	0.3652	1.0000	1.0000	1.0000
February	0.9945	0.9400	0.2240	1.0000	1.0000	1.0000
Harh	1.0000	0.9515	0.2181	1.0000	0.9950	1.0000
Januray	0.9914	0.7934	0.3272	0.9957	0.9957	0.9871
Jeth	0.9949	0.9024	0.3067	1.0000	1.0000	0.9949
July	1.0000	0.7984	0.2680	0.9902	1.0000	0.9951
June	1.0000	0.9615	0.1846	1.0000	1.0000	0.9888
Katak	1.0000	0.8235	0.2275	1.0000	1.0000	1.0000
Magar	1.0000	0.9471	0.3623	1.0000	1.0000	1.0000
Magh	0.9953	0.8370	0.1560	1.0000	0.9907	0.9906
March	0.9949	0.8040	0.1672	1.0000	1.0000	0.9899
May	0.9381	0.5764	0.2540	0.9899	0.9801	0.9949
November	0.9948	0.9945	0.7222	0.9948	1.0000	1.0000
October	0.9949	0.9412	0.2736	0.9949	1.0000	0.9899
Phagun	0.9801	0.7714	0.1669	0.9950	1.0000	0.9949
Poh	1.0000	0.9275	0.3605	1.0000	1.0000	1.0000
Sawan	0.9952	0.8662	0.2805	0.9954	0.9954	0.9864
September	0.9949	0.9261	0.7222	1.0000	0.9949	0.9949
Vaisakh	0.9951	0.6309	0.1923	0.9808	0.9903	0.9951
Overall Precision	0.9939	0.8627	0.3235	0.9965	**0.9974**	0.9951

From Table 8, it is clear that the proposed CNN model has given the best results for overall precision on the text image dataset, when tested using the Adam optimizer. The value of the overall precision using the Adam optimizer is around 0.9974, which is the highest of the overall precision results of other optimizers.

The value of the worst result for overall precision is 0.3235, which is using the Adadelta optimizer as per Table 8.

5.3 Recall Results Analysis

The recall result of each class of Gurmukhi month’s name for different optimizers is presented in Table 9.

Table 9
Recall Results for Each Class for Different Optimizers

	SGD	**Adagrad**	**Adadelta**	**RMSprop**	**Adam**	**Nadam**
April	1.0000	0.9065	0.1215	1.0000	1.0000	1.0000
Assu	1.0000	0.9267	0.1099	1.0000	1.0000	0.9948
August	0.9906	0.9906	0.3443	1.0000	0.9906	0.9953
Bhado	1.0000	0.7576	0.0202	1.0000	1.0000	0.9899
Chet	0.9890	0.4862	0.3370	0.9834	0.9945	0.9890
December	0.9900	0.9204	0.2090	1.0000	0.9950	0.9950
February	0.9891	0.7705	0.4590	0.9945	0.9945	0.9891
Harh	1.0000	0.9751	0.4677	0.9900	1.0000	0.9950
Januray	1.0000	0.9348	0.2304	0.9957	1.0000	1.0000
Jeth	0.9849	0.7437	0.2513	0.9849	0.9899	0.9849
July	0.9901	0.9754	0.3300	0.9951	0.9951	0.9951
June	1.0000	0.9831	0.6067	0.9888	1.0000	0.9944
Katak	1.0000	0.8235	0.2032	1.0000	0.9947	0.9947
Magar	1.0000	0.8483	0.1185	1.0000	1.0000	0.9905
Magh	0.9953	0.8962	0.2877	0.9906	1.0000	0.9953
March	1.0000	0.8163	0.2806	1.0000	1.0000	1.0000
May	1.0000	0.8426	0.2437	1.0000	1.0000	0.9949
November	1.0000	0.9427	0.2708	1.0000	1.0000	1.0000
October	0.9949	0.6531	0.1480	1.0000	0.9949	1.0000
Phagun	0.9949	0.8182	0.5455	1.0000	1.0000	0.9899
Poh	0.9950	0.8861	0.1535	1.0000	1.0000	1.0000
Sawan	0.9495	0.6239	0.1055	0.9954	0.9862	1.0000
September	1.0000	0.9641	0.0667	1.0000	1.0000	1.0000
Vaisakh	0.9902	0.7171	0.1220	0.9951	1.0000	0.9902
Overall Recall	0.9939	0.8418	0.2514	0.9964	**0.9973**	0.9949

As per Table 9, 0.9973 is the highest overall recall result for the proposed CNN model using Adam optimizer. This is the highest value for the overall recall among the overall recall results obtained by other optimizers.

On the other hand, the worst result for overall recall has been obtained using the Adadelta optimizer, whose value is 0.2514.

Finally, as per the result analysis of F1 score, precision and recall, it is concluded that the Adam optimizer on proposed CNN model outperformed other optimizers.

6. Results Analysis With Best Optimizers At Different Epochs

It has been proved in the previous sections that the Adam optimizer on the proposed CNN model has outperformed the other optimizers for Gurumukhi month's name image classification.

Furthermore, in this section, an experiment has been performed using the Adam optimizer on a proposed CNN model at different numbers of epochs of 100 and 40. The experimentation results of the Adam optimizer on the proposed model at different numbers of epochs in terms of training accuracy, validation accuracy, training loss and validation losses are presented in Figs. 5.

As depicted in Figs. 5, the proposed CNN model with Adam optimizer has achieved the best results in terms of highest training and validation accuracy at 100 epoch and 40 batch size, which is highlighted in blue.

7. Conclusion

In this research article, the architecture of the CNN model is designed and proposed for classification of Gurumukhi month's name recognition. The CNN architecture is designed with five convolutional, three polling layers, and one flatten layer and one dense layer. Experimentation was performed for the performance assessment of various deep learning optimizers in terms of validation accuracy, computational time, F1 score, precision, and recall on the proposed CNN model. The result showed that the proposed CNN model on 100 epochs and 40 batch size with Adam optimizers outperformed other optimizers.

Declarations

Compliance with Ethical Standards Statements

Ethical approval: Ethical approval is not applicable to manuscript.

Funding: The authors received no specific funding for this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

Informed Consent: Informed Consent is not applicable to manuscript.

Authorship Contributions

Tajinder Pal Singh -idea for the article, performed the literature search, prepared the dataset, design the CNN model, conceived the experiment, wrote the Paper.
Sheifali Gupta and Meenu Garg – Conceptualization, drafted and critically revised the work, formal analysis, analyzing and interpretation of data, supervision.
Deepika Koundal, Atef Zaguia - Formal analysis, drafted and critically revised the work, supervision.

References

Dhir, R. (2010). Moment based invariant feature extraction techniques for bilingual character recognition. 2010 2nd International Conference on Education Technology and Computer, V4-80-V4-84. https://doi.org/10.1109/ICETC.2010.5529730

Garg, A., Jindal, M. K., & Singh, A. (2019). Degraded offline handwritten Gurmukhi character recognition: Study of various features and classifiers. International Journal of Information Technology. https://doi.org/10.1007/s41870-019-00399-3

Jindal, U., Gupta, S., Jain, V., & Paprzycki, M. (2020). Offline Handwritten Gurumukhi Character Recognition System Using Deep Learning. In L. C. Jain, M. Virvou, V. Piuri, & V. E. Balas (Eds.), Advances in Bioinformatics, Multimedia, and Electronics Circuits and Signals (Vol. 1064, pp. 121–133). Springer Singapore. https://doi.org/10.1007/978-981-15-0339-9_11

Kumar, M., Jindal, M. K., & Sharma, R. K. (2011). Classification of characters and grading writers in offline handwritten Gurmukhi script. 2011 International Conference on Image Information Processing, 1–4. https://doi.org/10.1109/ICIIP.2011.6108859

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(1), 137–143. https://doi.org/10.1007/s40010-016-0284-y

Kumar, M., Jindal, M. K., Sharma, R. K., & Jindal, S. R. (2018). A novel framework for writer identification based on pre-segmented Gurmukhi characters. Sādhanā, 43(12), 197. https://doi.org/10.1007/s12046-018-0966-z

Kumar, M., Jindal, M. K., Sharma, R. K., & Jindal, S. R. (2020). Performance evaluation of classifiers for the recognition of offline handwritten Gurmukhi characters and numerals: A study. Artificial Intelligence Review, 53(3), 2075–2097. https://doi.org/10.1007/s10462-019-09727-2

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(1), 43–56. https://doi.org/10.1007/s11063-018-9913-6

Kumar, M., Sharma, R. K., & Jindal, M. K. (2014). Efficient Feature Extraction Techniques for Offline Handwritten Gurmukhi Character Recognition. National Academy Science Letters, 37(4), 381–391. https://doi.org/10.1007/s40009-014-0253-4

Kumar, N., Gupta, S., & Pradesh, H. (2017). A novel handwritten Gurmukhi character recognition system based on deep neural networks. International Journal of Pure and Applied Mathematics, 117(21), 663-678..

Kumar, R., Sharma, R. K., & Sharma, A. (2015). Recognition of Multi-Stroke Based Online Handwritten Gurmukhi Aksharas. Proceedings of the National Academy of Sciences, India Section A: Physical Sciences, 85(1), 159–168. https://doi.org/10.1007/s40010-014-0183-z

Mahto, M. K., Bhatia, K., & Sharma, R. K. (2018). Robust Offline Gurmukhi Handwritten Character Recognition using Multilayer Histogram Oriented Gradient Features. International Journal of Computer Sciences and Engineering, 6(6), 915–925. https://doi.org/10.26438/ijcse/v6i6.915925

Sakshi, Garg, N. K., & Kumar, M. (2018). Writer Identification System for Handwritten Gurmukhi Characters: Study of Different Feature-Classifier Combinations. In N. Chaki, A. Cortesi, & N. Devarakonda (Eds.), Proceedings of International Conference on Computational Intelligence and Data Engineering (Vol. 9, pp. 125–131). Springer Singapore. https://doi.org/10.1007/978-981-10-6319-0_11

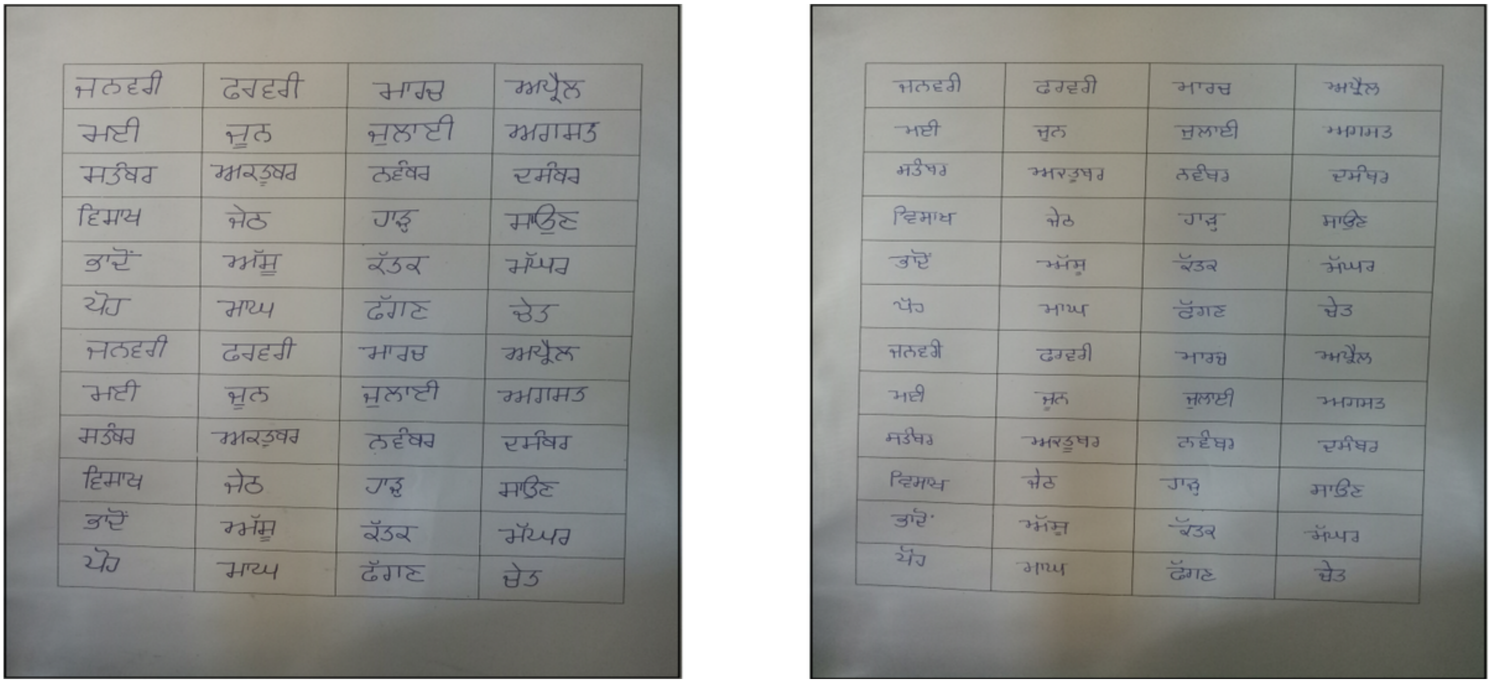
Sharma, A., Kumar, R., & Sharma, R. K. (2008). Online Handwritten Gurmukhi Character Recognition Using Elastic Matching. 2008 Congress on Image and Signal Processing, 391–396. https://doi.org/10.1109/CISP.2008.297

Sharma, A., Kumar, R., & Sharma, R. K. (2009). Rearrangement of Recognized Strokes in Online Handwritten Gurmukhi Words Recognition. 2009 10th International Conference on Document Analysis and Recognition, 1241–1245. https://doi.org/10.1109/ICDAR.2009.36

Singh, H., Sharma, R. K., & Singh, V. P. (2017). Efficient zone identification approach for the recognition of online handwritten Gurmukhi script. Neural Computing and Applications, 31(8), 3957–3968. https://doi.org/10.1007/s00521-017-3340-x

Verma, K., & Sharma, R. K. (2017). Recognition of online handwritten Gurmukhi characters based on zone and stroke identification. Sādhanā, 42(5), 701–712. https://doi.org/10.1007/s12046-017-0632-x

Figures



ਜਨਵਰੀ	ਫਰਵਰੀ	ਮਾਰਚ	ਅਪ੍ਰੈਲ
ਮਈ	ਜੂਨ	ਜੁਲਾਈ	ਅਗਸਤ
ਸਤੰਬਰ	ਅਕਤੂਬਰ	ਨਵੰਬਰ	ਦਸੰਬਰ
ਵਿਸਾਖ	ਜੇਠ	ਹਾੜ੍ਹ	ਸਾਉਣ
ਭਾਦੋਂ	ਅੱਸੂ	ਕੱਤਕ	ਮੱਘਰ
ਪੋਹ	ਮਾਘ	ਫੱਗਣ	ਚੇਤ
ਜਨਵਰੀ	ਫਰਵਰੀ	ਮਾਰਚ	ਅਪ੍ਰੈਲ
ਮਈ	ਜੂਨ	ਜੁਲਾਈ	ਅਗਸਤ
ਸਤੰਬਰ	ਅਕਤੂਬਰ	ਨਵੰਬਰ	ਦਸੰਬਰ
ਵਿਸਾਖ	ਜੇਠ	ਹਾੜ੍ਹ	ਸਾਉਣ
ਭਾਦੋਂ	ਅੱਸੂ	ਕੱਤਕ	ਮੱਘਰ
ਪੋਹ	ਮਾਘ	ਫੱਗਣ	ਚੇਤ

ਜਨਵਰੀ	ਫਰਵਰੀ	ਮਾਰਚ	ਅਪ੍ਰੈਲ
ਮਈ	ਜੂਨ	ਜੁਲਾਈ	ਅਗਸਤ
ਸਤੰਬਰ	ਅਕਤੂਬਰ	ਨਵੰਬਰ	ਦਸੰਬਰ
ਵਿਸਾਖ	ਜੇਠ	ਹਾੜ੍ਹ	ਸਾਉਣ
ਭਾਦੋਂ	ਅੱਸੂ	ਕੱਤਕ	ਮੱਘਰ
ਪੋਹ	ਮਾਘ	ਫੱਗਣ	ਚੇਤ
ਜਨਵਰੀ	ਫਰਵਰੀ	ਮਾਰਚ	ਅਪ੍ਰੈਲ
ਮਈ	ਜੂਨ	ਜੁਲਾਈ	ਅਗਸਤ
ਸਤੰਬਰ	ਅਕਤੂਬਰ	ਨਵੰਬਰ	ਦਸੰਬਰ
ਵਿਸਾਖ	ਜੇਠ	ਹਾੜ੍ਹ	ਸਾਉਣ
ਭਾਦੋਂ	ਅੱਸੂ	ਕੱਤਕ	ਮੱਘਰ
ਪੋਹ	ਮਾਘ	ਫੱਗਣ	ਚੇਤ

Figure 1

(a) Sample Sheet from Writer 1 (b) Sample Sheet from Writer 2

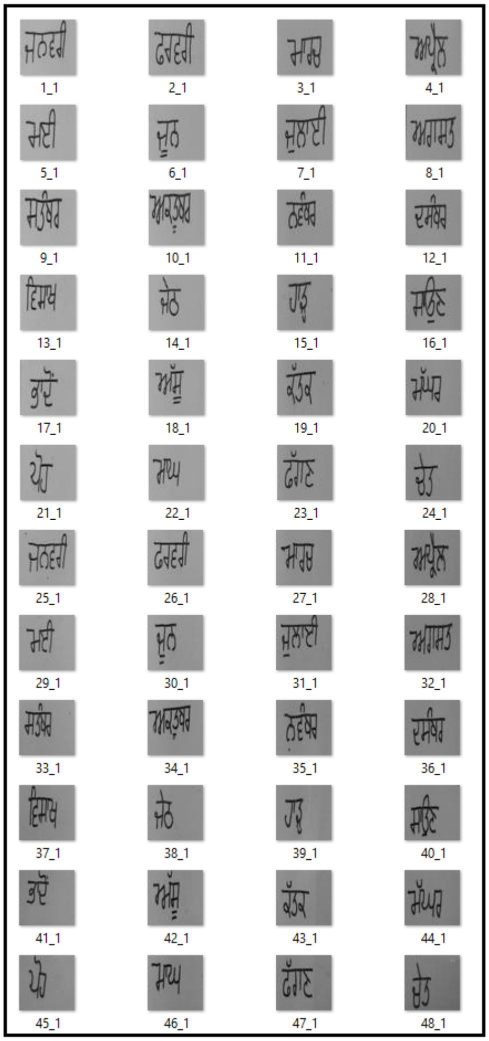


Figure 2

Cropped Images

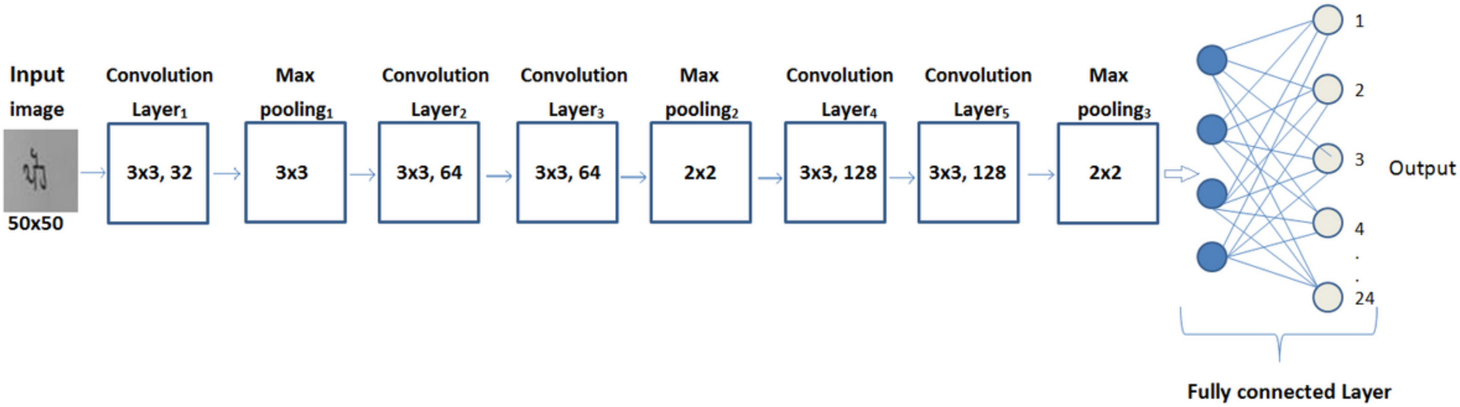


Figure 3

Proposed CNN Model Architecture

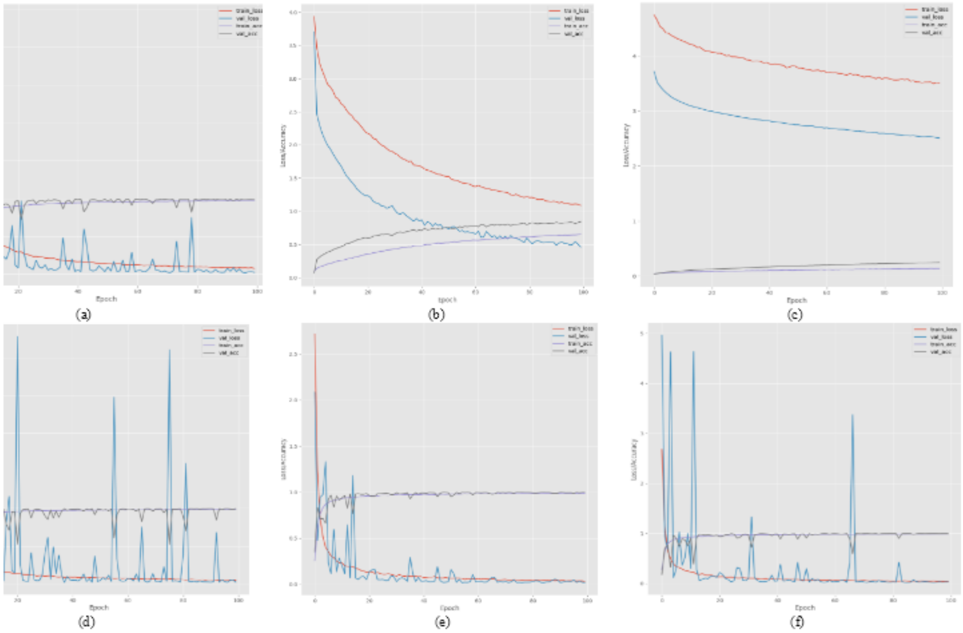


Figure 4

Training and Validation Accuracy/Loss Graph for Different Optimizers (a) SGD (b) Adagrad (c) Adadelta (d) RMSprop (e) Adam (f) Nadam

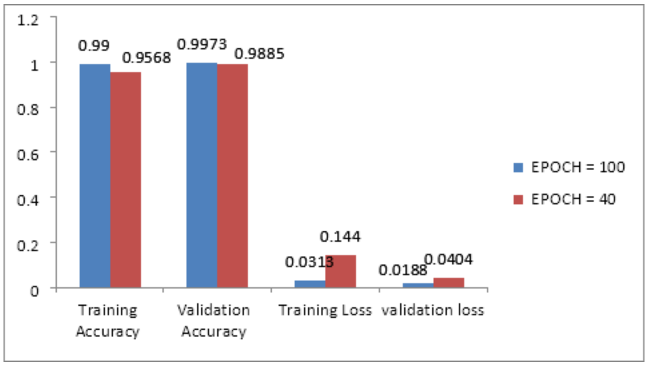


Figure 5

Performance of Proposed Model with Adam optimizer at 100 and 40 Epochs with Batch size of 40