

## Research Article

# Transfer and Deep Learning-Based Gurmukhi Handwritten Word Classification Model

Tajinder Pal Singh <sup>1</sup>, Sheifali Gupta <sup>1</sup>, Meenu Garg <sup>1</sup>, Amit Verma <sup>2</sup>, V. V. Hung,<sup>3</sup>  
H. H. Thien <sup>3</sup> and Md Khairul Islam <sup>4</sup>

<sup>1</sup>Chitkara University Institute of Engineering and Technology, Chitkara University, Punjab, India

<sup>2</sup>University of Petroleum & Energy Studies, Dehradun, India

<sup>3</sup>Faculty of Information Technology, Ho Chi Minh City Open University, Ho Chi Minh City, Vietnam

<sup>4</sup>Department of Information & Communication Technology, Islamic University, Kushtia 7003, Bangladesh

Correspondence should be addressed to Amit Verma; [amit.uptu2006@gmail.com](mailto:amit.uptu2006@gmail.com) and Md Khairul Islam; [mdkito51@gmail.com](mailto:mdkito51@gmail.com)

Received 16 August 2022; Revised 19 September 2022; Accepted 28 September 2022; Published 3 May 2023

Academic Editor: Savita Gupta

Copyright © 2023 Tajinder Pal Singh et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The world is having a vast collection of text with abandon of knowledge. However, it is a difficult and time-taking process to manually read and recognize the text written in numerous regional scripts. The task becomes more critical with Gurmukhi script due to complex structure of characters motivated from the challenges in designing an error-free and accurate classification model of Gurmukhi characters. In this paper, the author has customized the convolutional neural network model to classify handwritten Gurmukhi words. Furthermore, dataset has been prepared with 24000 handwritten Gurmukhi word images with 12 classes representing the month's names. The dataset has been collected from 500 users of heterogeneous profession and age group. The dataset has been simulated using the proposed CNN model as well as various pretrained models named as ResNet 50, VGG19, and VGG16 at 100 epochs and 40 batch sizes. The proposed CNN model has obtained the best accuracy value of 0.9973, whereas the ResNet50 model has obtained the accuracy of 0.4015, VGG19 has obtained the accuracy of 0.7758, and the VGG16 model has obtained value accuracy of 0.8056. With the current accuracy rate, noncomplex architectural pattern, and prowess gained through learning using different writing styles, the proposed CNN model will be of great benefit to the researchers working in this area to use it in other ImageNet-based classification problems.

## 1. Introduction

In the application of natural language processing, handwritten text recognition is used on records for data analysis and recognition. Through data analysis and recognition, it provides an interface for the improvement of communication between humans and computers. Nowadays, this process become more popular for regional applications as most of the information has got digitized everywhere. However, as compared to non-Indian scripts, handwritten text recognition for Indian regional scripts has not achieved expected perfection due to the complex structure of scripts such as Gurmukhi. Hence, it remains an active research area till now.

Generally, handwritten text recognition systems are developed based on an analytical word recognition approach or a holistic word recognition approach. Handwritten text recognition based on analytical word recognition performs character level segmentation of a word before text recognition. On the other hand, text recognition based on holistic word recognition is a segmentation-free approach in which the whole word is recognized at once using its contour or shape information. Hence, holistic word recognition technique is very common for error-free results on documents comprising of overlapped or touched characters as there is no need of word segmentation. In the present work, the authors have also employed a holistic approach to handwritten Gurmukhi word recognition.

Furthermore, handwritten text recognition systems that have been developed are based on machine learning algorithms or deep learning algorithms. Mostly, machine learning algorithms based on handwritten text recognition systems approach to manual features' extraction for text recognition, in which low-level or high-level character-level features have been extracted for interpreting the text accurately. For example, both [1, 2] have employed machine learning to recognize the online handwritten Gurmukhi characters using manual extracted low-level and high-level features of Gurmukhi characters and have achieved a 90.08% recognition efficiency. In order to extract the moment-based invariant features of 52 bilingual (Roman and Gurmukhi) characters for recognition, Dhir [3] applied various moments such as Zernike moments, pseudo-Zernike moments, and orthogonal Fourier–Mellin moments. Out of these, pseudo-Zernike moments outperformed the other moments. Kumar et al. [4] employed machine learning to compare the handwriting of various writers in Gurmukhi. For this, the authors have prepared a feature set of manually extracted zoning, directional, and diagonal features. The classification results of this article show that writer 6 has achieved the best accuracy. Kumar et al. [5] employed machine learning-based classification techniques for text recognition on manual extracted features of Gurmukhi characters that are parabola curve fitting-based features and power curve fitting-based features. The author has achieved an accuracy rate of up to 98.10% using this approach in the recognition of offline handwritten Gurmukhi characters. Both [6, 7] have developed online handwritten Gurmukhi character recognition systems based on various machine learning-based classification techniques and 64-points feature set extracted from online handwritten Gurmukhi characters. Similarly, Kumar et al. [8] have also developed an offline handwritten Gurmukhi character recognition system using machine learning classifiers and various transformation techniques such as discrete wavelet transformations (DWT2), discrete cosine transformations (DCT2), fast Fourier transformations, and fan beam transformations that were used for Gurmukhi character's feature extraction.

Singh et al. [9] have used machine learning for online handwritten Gurmukhi script recognition based on points feature, discrete Fourier transformation features, and directional features. The proposed approach has achieved recognition accuracy of 97.07%. Both [10, 11], for developing writer identification systems in Gurmukhi text, have used machine learning-based classification techniques and feature sets comprising of zoning, transitions, peak extent, centroid, parabola curve fitting, and power curve fitting-based features. An accuracy of 89.85% and an average of 81.75% have been achieved using this approach. Kumar et al. [12] have proposed an optical character recognition system that has been developed using machine learning classifiers and manual extracted features such as zoning, discrete cosine transformations, and gradient feature extraction. The proposed optical character recognition system has been trained on the 18th to 20th century's mediaeval handwritten documents and achieved a 95.91% accuracy rate in the

recognition of works written by distinct writers. On degraded handwritten Gurmukhi characters, Garg et al. [13] have achieved recognition accuracy of 96.03% using machine learning classifiers and zoning, diagonal, shadow, and peak extent-based features. In order to evaluate the performance of character recognition systems based on machine learning algorithms, Kumar et al. [14] have experimented on a dataset of handwritten Gurmukhi characters and numerals. The system's accuracy of up to 87.9% has been achieved when peak extent diagonal and centroid kinds of features were extracted from characters and numbers.

In contrast to handwritten text recognition systems based on machine learning algorithms, a handwritten text recognition system based on deep learning can be developed through customized convolution neural networks or pre-trained transfer learning models. Both customized convolution neural networks and pretrained models work on automatically extracting features from handwritten text. Specifically, pretrained models, in the context of deep learning, are the methods that use the features that have been learned by a network on a given problem to solve a different set of problems in the same domain. Previously, various authors have used pretrained models for handwritten text recognition. For example, Ganji et al. [15] employed a transfer learning model named "VGG16" to recognize the multivariant handwritten Telugu character. The proposed approach by the authors in this article is the alternative to the already existing optical character recognition systems, which were not capable of recognizing the variant handwritten Telugu characters due to the availability of a limited dataset. A maximum accuracy of 92% has been achieved using this method. In order to recognize the handwritten words using a holistic approach, Pramanik and Bag [16] used AlexNet, VGG-16, VGG-19, ResNet50, and GoogleNet on a dataset of Bangla city names. The author has achieved a maximum accuracy of 98.86% with ResNet50 in this experiment. Similarly, Pramanik et al. [17] have performed experimentation for the recognition of Bangla handwritten pin codes on postal letters as a part of practical application using an already available pretrained CNN model on various Indic scripts, including Bangla, Devanagari, Oriya, and Telugu. When tested on 28 postcards and 168 total digits, the pretrained models, AlexNet and VGG16, achieved maximum efficiency of 94.26% and 92.21%, respectively, in this experimentation. As the pretrained model has been used for developing a handwritten text recognition system based on deep learning, researchers have also employed customized convolution neural networks to develop the same. For example, [18] customized a convolution neural network with two convolutional layers and two max-pooling layers to recognize 3500 Gurmukhi characters. The results show that the customized convolution neural network in this article has achieved 98.32% accuracy on the training set and 74.66% on the testing set data. To effectively recognize the offline handwritten text, a sequence-to-sequence method based on CNN-RNN models has been developed by [19]. The suggested model demonstrated competitive word recognition accuracy when tested on the IAM and RIMES handwritten

datasets. Furthermore, in order to recognize online Gurmukhi words, [20] customized a convolution neural network with seven convolutional layers and three maximum pooling layers that gives 97% accuracy. Similarly, [21] have a customized convolution neural network with 1 convolution and three capsule layers, [22] customized the binary neural network, [23] customized the deep neural network based on ResNet 50 called CORNet, and [24] customized the convolutional neural network with 19 layers including 6 convolutional, 3 max-pooling, 4 dropout layers, 3 batch normalization layers, 1 flatten layer, and 2 dense layers.

Pretrained models somehow give numerous benefits of using it, such as saving computational time and being very useful on small datasets, but it may face some critical issues for solving different problems in the same domain where it is trained, such as negative transfer and inaccuracy in identifying decision boundaries among multiple classes in the dataset of the target domain. This will result in making it unsuitable for real-time applications such as automated month's name recognition. Hence, it is recommended to build a custom convolution neural network whose learning will be initialized from scratch for the purpose of effective performance in such applications of automated text recognition systems.

With the objective of proposing a design for an automated month's name recognition system in the Gurmukhi language, the authors have customized a convolutional neural network with 5 convolutional and 3 pooling layers from scratch. For the present classification problem, the weights of a customized convolution neural network are initialized from the beginning and updated through learning the problem, irrespective of utilizing the frozen weights of pretrained models. This leads to better identification of decision boundaries among the 24 classes on a given dataset in the target domain using a customized convolution neural network. As a result of this, the proposed CNN model has shown better performance than the pretrained model on a custom dataset in terms of training results as well as results for confusion matrix parameters.

The major contribution in terms of novelty of the present research work in the target domain has been well explained in the following points:

- (1) A convolutional neural network has been customized from scratch with 5 convolutional and 3 pooling layers for an automated month's name recognition system in the Gurmukhi language for regional applications. The proposed CNN design in term of its architectural pattern is less complex and less prone to over fitting when compared to various transfer learning models used for image classification task in the given context. Hence, the proposed CNN model is suitable for portable automated text recognition systems.
- (2) The performance of customized convolutional neural network for handwritten text recognition system has been tested on custom dataset of handwritten words that has been prepared from 500 distinct writers from various professions and age

groups. Hence, the proposed CNN model is a prowess model through the learning of different writing styles.

- (3) Furthermore, it has been observed from the result analysis that the customized convolutional neural network on handwritten Gurmukhi month's name dataset possessed an accuracy rate of up to 99.73%, which is more than using various transfer learning models named as ResNet50, VGG 19, and VGG16, respectively, on the same dataset. As a result, the proposed CNN model's learning weights can be exploited for the use in other ImageNet-based classification problems in the given context.

The present article has been divided into various sections. Section 1 focuses on the introduction and literature review of the article. Section 2 describes the materials and methods used, while conducting the experiment, and also details the complete description of the proposed CNN model's architecture. In Section 3 of this article, results and analysis of the proposed experimentation work have been presented in terms of results obtained at different training parameters and a comparative analysis performed among the various models that have been chosen in this work. Finally, Section 4 has concluded the overall results of various experiments performed for the performance assessment of the proposed CNN model in the classification of Gurmukhi month's name images and its comparison to other three pretrained models.

## 2. Materials and Methods

This section provides a detailed description of the step-by-step approach of dataset preparation and the methodology used for the classification of handwritten word datasets.

*2.1. Dataset.* The dataset for the present research work has been prepared for 24,000 handwritten word images belonging to 24 different classes of Gurmukhi month's names that represent the handwriting style variation among 500 distinct writers from different professions and age groups.

The various steps for dataset preparation have been shown in Figure 1.

The step-by-step approach of dataset preparation has been described by the following points:

- (1) In the first step, an offline handwritten Gurmukhi month's name dataset was created on an A4 sheet of paper that contains 48 blocks of the same size for writing the handwritten words. A single writer wrote the names of the 24 classes of the Gurmukhi months two times on an A4 sheet and gave 48 handwritten words on a single piece of paper. Similarly, the dataset for handwritten words of 24 classes of Gurmukhi month's name has been collected on 500 A4-sized sheets from 500 distinct writers, resulting in the formation of 24,000 words in the Gurmukhi month's name dataset. This step helps in

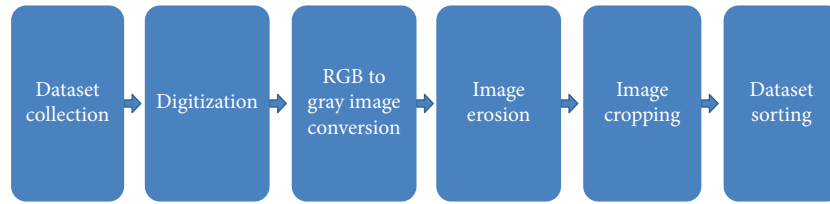


FIGURE 1: Steps to dataset preparation.

incorporating different writing styles into the dataset, which will help to process the proposed CNN model through learning it.

- (2) All 500 A4-sized sheets collected from 500 distinct writers have been digitized in the second step using an OPPO F1s smart phone camera. Each digitized image of sample sheet is of size  $1024 \times 786$  pixels. Using this step called digitization, offline hand-written word recognition has become possible and easy.
- (3) In the third step of dataset preparation, colour to gray image conversion has been performed on digitized dataset using MATLAB operation “rgb2gray.” A conversion of colour to gray image has reduced the data size as well as the computation cost, which results in fast learning and recognition rate.
- (4) The dataset preparation in the fourth step has performed eroding on all gray scale images using “imerode” function of MATLAB software. Image eroding helps in retrieving the valuable information from digitized text data.
- (5) In the fifth step of dataset preparation, the dataset of eroded images has been cropped into word images and collected 48-word images from each sample erode image. This operation has been performed using “imcrop” function of MATLAB software. This operation helps in sorting the dataset classwise.
- (6) In the sixth step of dataset preparation, class name-wise dataset sorting has been performed on cropped images. For each class in the 24 classes of Gurmukhi month’s name, cropped images have been sorted into the class’s respective folders. This results in a collection of 1000 images for each class name in the Gurmukhi month’s name dataset. Using this operation, distinction has been incorporated among each image of the different classes of the Gurmukhi month’s name.

The results of dataset preparation are shown in Figure 2.

The customized dataset for this experiment was divided into training and testing sets in an 80 : 20 ratio. That means a total of 19,200 labeled samples from the Gurmukhi handwritten words dataset have been used as a training set and the remaining 4,800 labeled samples have been used as a testing set. The detailed description of dataset as well as the training set and testing set samples selected from each class of Gurmukhi month’s name in the present work is shown in Table 1.

*2.2. Proposed CNN Model.* This section will illustrate the design and development of a proposed CNN model that is built on alternating convolution layers, normalization layers, max-pooling layers, and dropouts, followed by some fully connected layers. In general, the CNN is a multilayer neural network that stores numerous trainable weights and biases, while training using forward and backpropagation algorithms. A CNN model is constructed through automatic feature extractors and trained classifiers, which help in efficiently learning complex or high-dimensional data to solve image classification problems. For the present research work, the proposed sequential CNN model has been composed of 5 convolutional layers, 3 max-pooling layers, and 1 fully connected layer. The architectural diagram of the proposed CNN model is shown in Figure 3.

*2.3. Proposed CNN’s Layer Description.* The proposed CNN model’s layers have been classified into two types, namely, primary layers and secondary layers. The primary layers of the proposed CNN model include convolution layers, activation layers, pooling layers, flattening layers, and dense layers, which are the main layers used in the CNN. On the other hand, the secondary layers of the proposed CNN model are the optional layers that have been added to improve resistance to over-fitting of CNN networks, which is named as dropout layers and batch normalization layers. The details descriptions of CNN model’s layers and learning parameters are presented in Table 2.

*2.4. Automatic Feature Extraction.* For automatic extraction of features from an image, three layers in the proposed CNN model play a crucial role. These layers are the convolution layers, the max-pooling layer, and batch normalization. As discussed earlier, the proposed CNN model has been built using five convolutional layers, three max-pooling layers, and one layer of batch normalization after each convolutional operation. All five convolution layers of the proposed CNN model have been designed with the same size of filter ( $3 \times 3$ ) but with a different filter count in each convolution layer. For example, the first convolutional layer of the proposed CNN model has 32 weight filters, the second and third convolutional layers have 64 weight filters, and the fourth and fifth convolutional layers have 128 weight filters. A batch normalization layer has been added after each and every convolutional layer, as it is required to keep the output away from the saturation region using the mean and variance. Out of three max-pooling layers, the first layer of

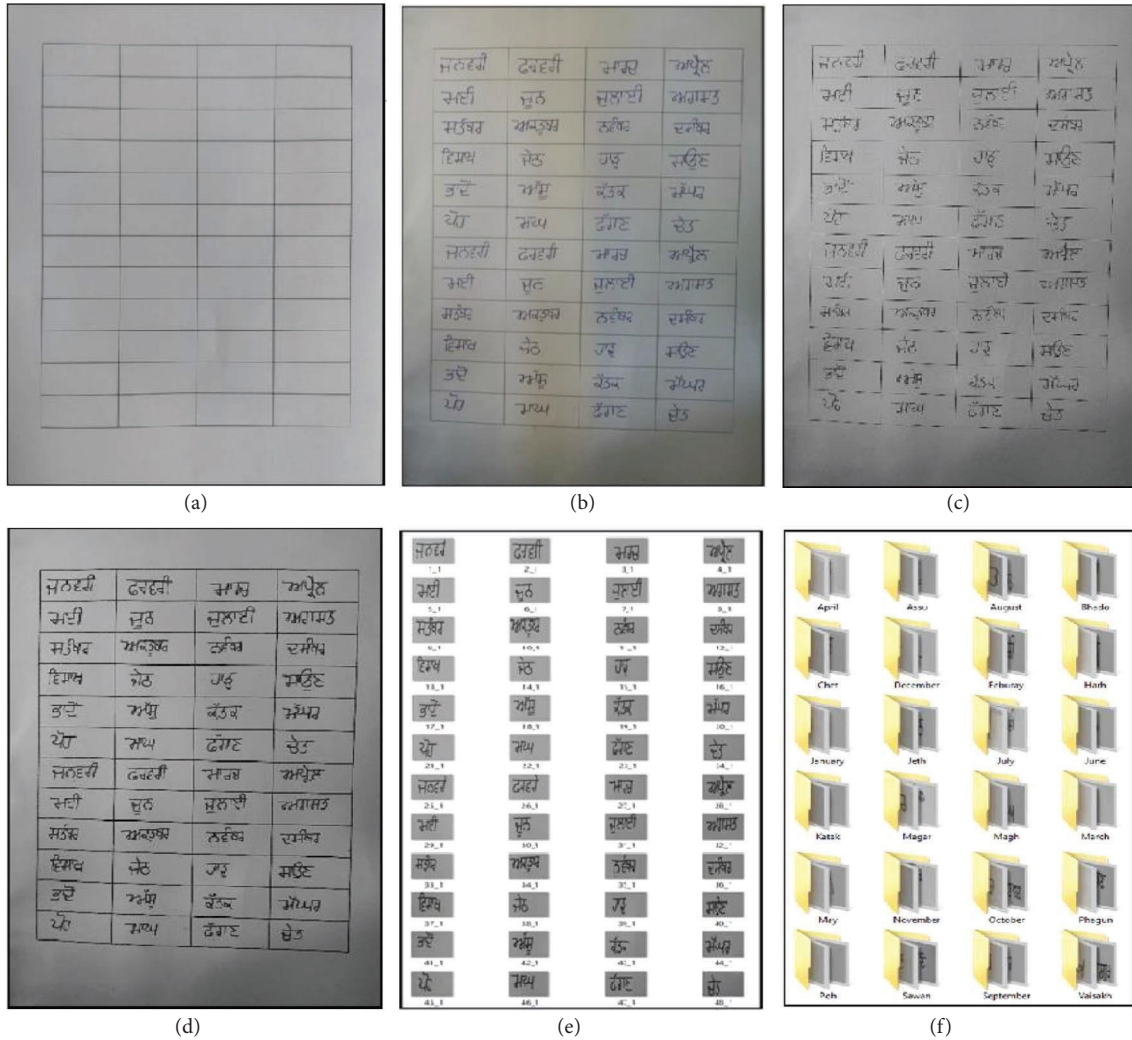


FIGURE 2: (a) Blank sample sheet. (b) Sample sheet from writer 1. (c) Image converted from RGB to Gray. (d) Eroded image. (e) Cropped images of Gurmukhi Month's name. (f) Folders containing sorted dataset.

pooling comprises a filter size of  $3 \times 3$  and the second and third pooling layers comprise a filter size of  $3 \times 3$ . In order to reduce over fitting, a dropout layer has also been used as the next layer, and it has been configured to randomly and temporarily remove 25% of the neurons. These layers all together form an automatic feature extractor for the proposed CNN model.

**2.5. Classification.** After feature extraction, the next step is data classification using the model. For this, initially flattening layer transforms the two-dimensional matrix of the features extracted through convolutional operation into a single vector and feeds it to the first fully connected layer of the model. The fully connected layer comprises 1024 neurons and a ReLu activation function. After this, a layer of batch normalization and a dropout layer of 0.5 have been added. Finally, a second fully connected layer has been added to the network with softmax as an activation function in order to classify handwritten word data into 24 classes.

**2.6. Experimental Setup.** For obtaining the results of simulation of Gurmukhi month's name classification using self-created handwritten word dataset, an experiment has been performed on Tesla K80 GPU that is virtually available on Google Colab, under the constrained of its limited use to maximum 12 hours for continuous training. The maximum limit to use the GPU also varies depending upon the traffic on GPU at that time [21, 24]. Along with this, the memory of other hardware RAM and ROM used for this experiment is 12.69 GB and 107.72 GB, respectively.

Furthermore, because of the availability of very useful libraries in Python [21], proposed CNN model's architectures have been built using Keras with a TensorFlow backend.

Other than this, for training the proposed CNN model, selected parameters are presented in Table 3.

For selecting an appropriate deep optimizer for the proposed CNN model's simulation on the Gurmukhi handwritten word dataset with an effective accuracy rate, performance compromise has been made among different

TABLE 1: Detail description of dataset with the number of training and testing samples selected for each class.

	Class 1	Class 2	Class 3	Class 4	Class 5	Class 6	Class 7	Class 8	Class 9	Class 10	Class 11	Class 12	Class 13	Class 14	Class 15	Class 16	Class 17	Class 18	Class 19	Class 20	Class 21	Class 22	Class 23	Class 24
Class name in English	Vaisakh	Jeth	Harh	Sawan	Bhaddo	Asu	Katak	Magar	Poh	Magh	Phagun	Chet	January	February	March	April	May	June	July	August	September	October	November	December
Class name in Gurumukhi	ਵੈਸਾਖ	ਜੇਠ	ਹਾਠ	ਸਾਵਣ	ਭਾਦੋ	ਅਸੂ	ਕਾਕ	ਮਾਗਰ	ਪੋਹ	ਮਾਘ	ਫਾਗਣ	ਚੈਤ	ਜਨਵਰੀ	ਫਰਵਰੀ	ਮਾਰਚ	ਅਪਰੈਲ	ਮਈ	ਜੂਨ	ਜੁਲਾਈ	ਅਗਸਟ	ਸਤੰਬਰ	ਅਕਤੂਬਰ	ਨਵੰਬਰ	ਦਸੰਬਰ
Duration	14 April to 14 May	15 May to 14 June	15 June to 15 July	16 July to 15 August	16 August to 14 September	15 September to 14 October	15 October to 13 November	14 November to 13 December	14 December to 12 January	13 January to 11 February	12 February to 13 March	14 March to 13 April	1 January to 13 February	1 February to 28/29 February	1 March to 31 March	1 April to 30 April	1 May to 31 May	1 June to 30 June	1 July to 31 July	1 August to 31 August	1 September to 30 September	1 October to 31 October	1 November to 30 November	1 December to 31 December
Number of training samples	795	801	799	782	802	809	813	789	798	788	802	819	770	817	804	786	803	822	797	788	805	804	808	799
Number of testing samples	205	199	201	218	198	191	187	211	202	212	198	181	230	183	196	214	197	178	203	212	195	196	192	201



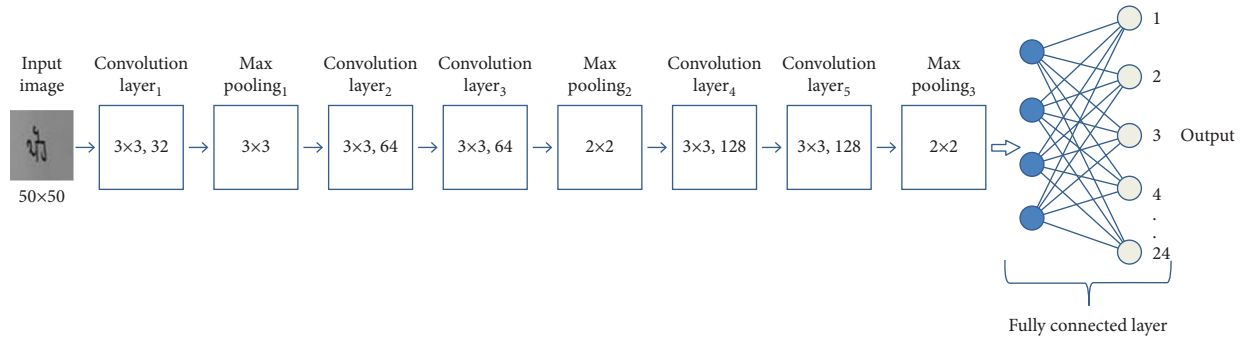


FIGURE 3: Proposed CNN model’s architecture.

TABLE 2: Layer’s description of the proposed CNN model.

Model: “Sequential” layer (type)	Output shape	Param#
conv2d (Conv2D)	(None, 50, 50, 32)	320
activation (Activation)	(None, 50, 50, 32)	0
batch normalization (Batch Normalization)	(None, 50, 50, 32)	128
max_pooling2d (MaxPooling2D)	(None, 16, 16, 32)	0
dropout (Dropout)	(None, 16, 16, 32)	0
conv2d_1 (Conv2D)	(None, 16, 16, 64)	18496
activation_1 (Activation)	(None, 16, 16, 64)	0
batch_normalization_1 (Batch Normalization)	(None, 16, 16, 64)	256
conv2d_2 (Conv2D)	(None, 16, 16, 64)	36928
activation_2 (Activation)	(None, 16, 16, 64)	0
batch_normalization_2 (Batch Normalization)	(None, 16, 16, 64)	256
max_pooling2d_1 (Max-Pooling 2D)	(None, 8, 8, 64)	0
dropout_1 (Dropout)	(None, 8, 8, 64)	0
conv2d_3 (Conv2D)	(None, 8, 8, 128)	73856
activation_3 (Activation)	(None, 8, 8, 128)	0
batch_normalization_3 (Batch Normalization)	(None, 8, 8, 128)	512
conv2d_4 (Conv2D)	(None, 8, 8, 128)	147584
activation_4 (Activation)	(None, 8, 8, 128)	0
batch_normalization_4 (Batch Normalization)	(None, 8, 8, 128)	512
max_pooling2d_2 (MaxPooling2D)	(None, 4, 4, 128)	0
dropout_2 (Dropout)	(None, 4, 4, 128)	0
flatten (Flatten)	(None, 2048)	0
dense (Dense)	(None, 1024)	2098176
activation_5 (Activation)	(None, 1024)	0
batch_normalization_5 (Batch Normalization)	(None, 1024)	4096
dropout_3 (Dropout)	(None, 1024)	0
dense_1 (Dense)	(None, 24)	24600
activation_6 (Activation)	(None, 24)	0

Total params: 2,405,720, trainable params: 2,402,840, and nontrainable params: 2,880.

optimizers such as stochastic gradient descent (SGD), Adagrad, Adadelata, RMSprop, Nadam, and Adam, and it has been observed that the proposed CNN model with the Adam optimizer has given the best accuracy, where the accuracy rate with other optimizers has been observed to be less than that of Adam.

Accuracy serves as the most fundamental and frequently used metric for measuring CNN models. However, F1 score, precision, and recall are all necessary to evaluate the mode’s quality. As a result, for the performance assessment of the proposed CNN model, all of these parameters have been incorporated.

While selecting an appropriate learning rate, the proposed CNN model has been initially simulated at the highest learning rate, but the best result has been observed using 0.0001 learning rate in terms of accuracy.

### 3. Results and Analysis

This section will discuss the training parameters results of the proposed CNN model and the pretrained model in detail. Furthermore, this section will also perform a comparative analysis between the proposed CNN model and the pretrained transfer learning models (ResNet, VGG19, and VGG16) in terms of accuracy as well as various confusion

TABLE 3: Training parameters of the proposed model.

Adam Optimizer's specification	Learning rate (LR)	Loss function	Matrix	Number of epochs	Batch size (BS)
Learning rate = $1E-03$ , beta1 = 0.9, beta2 = 0.999, epsilon = $1e-07$ , and decay = learning rate/epochs	0.0001	Categorical cross entropy	Accuracy	100	20



matrix parameters in order to examine the performance of the proposed CNN model thoroughly.

*3.1. Training Parameters' Results of the Proposed CNN Model.* In order to examine the performance of the proposed CNN model, the self-prepared handwritten word dataset has been simulated using the proposed CNN model at 100 and 40 epochs using 20 and 40 batch sizes, respectively. The various training parameters results of the proposed CNN model at different epochs and batch sizes are presented in Table 4.

From Table 4, it has been analyzed that the proposed CNN model at 100 epoch and 40 batch size gives the highest accuracy and minimum loss when simulated using a self-prepared handwritten word dataset. The values of the highest training and validation accuracy for the proposed CNN model on the handwritten word dataset are 99% and 99.73%, respectively.

On the other hand, the poor results have been analyzed for the proposed CNN model at 40 epochs using 20 batch sizes on the same dataset, whose values are 94.86% and 89.95% for training and validation accuracy, respectively.

The accuracy and loss graph for the proposed CNN model at different epochs (100 and 40) and batch sizes (20 and 40) is presented in Figure 4.

*3.2. Training Parameters' Results of ResNet 50.* The ResNet50 is designed with 48 convolutional layers that belong to four stages and four convolutional blocks of the model's architecture. The filter sizes that have been used for ResNet's convolutional and max-pooling layers are  $7 \times 7$  and  $3 \times 3$ . The complete architectural diagram of ResNet50 is presented in Figure 5.

The first stage of ResNet50 has 3 residual blocks and 3 layers in each block. Each layer in the first stage contains 64 or 256 feature maps of sizes of  $1 \times 1$  or  $3 \times 3$  for the convolution operation. The second stage of ResNet50 has also 3 residual blocks and each residual block has 4 layers of convolution. The numbers of the filters used to perform the operations are 128 or 512 in each layer. The model's third stage has 3 residual blocks, each with 6 layers. The numbers of filters that have been used in the third stage convolutional layers are either 256 or 1024. Finally, stage 4 of ResNet 50 consists of 3 residual blocks with 3 convolutional layers each. The sizes of the filters used to perform the operation are 512 and 2048. The last layer is the fully connected layer of the architecture that classifies handwritten word dataset into 24 classes.

The pretrained model ResNet 50 has been simulated on self-prepared handwritten dataset of Gurmukhi month's name at 100 and 40 epochs with 20 and 40 batch sizes, respectively. The training parameters' results obtained for this model is presented in Table 5.

From Table 5, it has been analyzed that the pretrained model ResNet 50 has performed the worst on the proposed handwritten words dataset of Gurmukhi month's name. The value of highest validation accuracy that has been obtained using this model is 40.15% at 100 epochs and 40 batch sizes. However, this accuracy result of ResNet 50 at 100 epochs and

40 batch sizes is comparatively higher than the performance of the model at 100 and 40 epochs using 20 batch sizes.

The accuracy loss graph of ResNet 50 at different epochs and batch sizes is presented in Figure 6.

*3.3. Training Parameters' Results of VGG 19.* The architectural diagram of VGG 19 is presented in Figure 7. This model has been designed with 19 total layers, of which 16 layers are convolutional layers and 3 layers are fully connected layers. Other than this, the model has also contained 5 max pool layers. The size of filter that has been used for operation in VGG 19 is of the same size  $3 \times 3$  for convolutional and pooling layers. The number of feature maps that has been used by each layer of VGG 19 is different and varies from 64 to 512. The last three layers are the fully connected layer of the architecture that classifies handwritten word dataset into 24 classes.

The pretrained model VGG19 has also been simulated on self-prepared handwritten dataset of Gurmukhi month's name at 100 and 40 epochs with 20 and 40 batch sizes, respectively. The training parameters' results obtained for this model is presented in Table 6.

It has been analyzed from Table 6 that the highest validation accuracy of pretrained mode VGG19 is 77.71% and 77.58%, respectively, when tested on 100 epochs with 20 and 40 batch sizes. Other than this, the least validation accuracy has been noticed for 40 epochs and 40 batch sizes, whose value is 73.17%. Hence, this model performed well on 20 and 40 batch sizes with 100 epochs in comparison to its performance of 40 epochs.

The VGG19's accuracy loss graph at different epochs and batch sizes has been presented in Figure 8.

*3.4. Training parameters' Results of VGG 16.* The VGG 16 model's constructional design is shown in Figure 9. The model comprises 16 layers in total, out of which 13 layers are convolutional layers and 3 layers are fully connected layers. Other than this, VGG16 also consists of five max-pooling layers. The filter size throughout all of the layers of the architecture is of same size, that is,  $3 \times 3$ . However, the count for the feature maps has varied from 64 to 512 from the layer first to the last layer of the model. The last three layers of the model are the fully connected layers that classify the handwritten word dataset into 24 classes.

The training parameters' results obtained for VGG16 when it is simulated on a self-prepared handwritten dataset of Gurmukhi month's name at 100 and 40 epochs with 20 and 40 batch sizes, respectively, have been presented in Table 7.

Table 6 depicted that the model's highest validation accuracy has been obtained at 100 epochs with 20 and 40 batch sizes, whose value is 81.38% and 80.56%, respectively. In contrast to this, the least accuracy for the model has been observed at 40 epochs and 40 batch sizes with an accuracy value of 75.63%. Hence, this model performed well on 20 and 40 batch sizes with 100 epochs in comparison to its performance of 40 epochs.

TABLE 4: Training parameters' results of the proposed CNN model at different epochs and batch sizes.

Results	Training parameters			
	Epoch = 100		Epoch = 40	
	Batch size 20	Batch size 40	Batch size 20	Batch size 40
Training accuracy	0.9878	0.99	0.9486	0.9568
Validation accuracy	0.9973	0.9973	0.8985	0.9885
Training loss	0.0432	0.0313	0.1580	0.1440
Validation loss	0.0225	0.0188	1.0888	0.0404

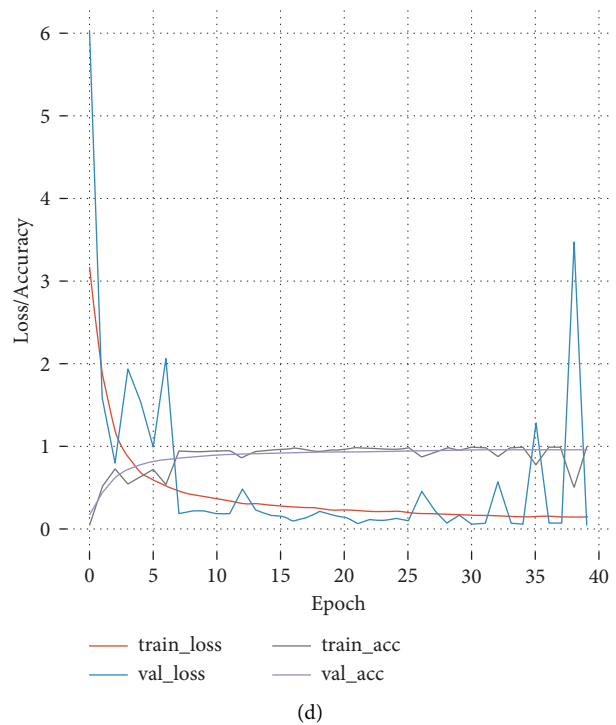
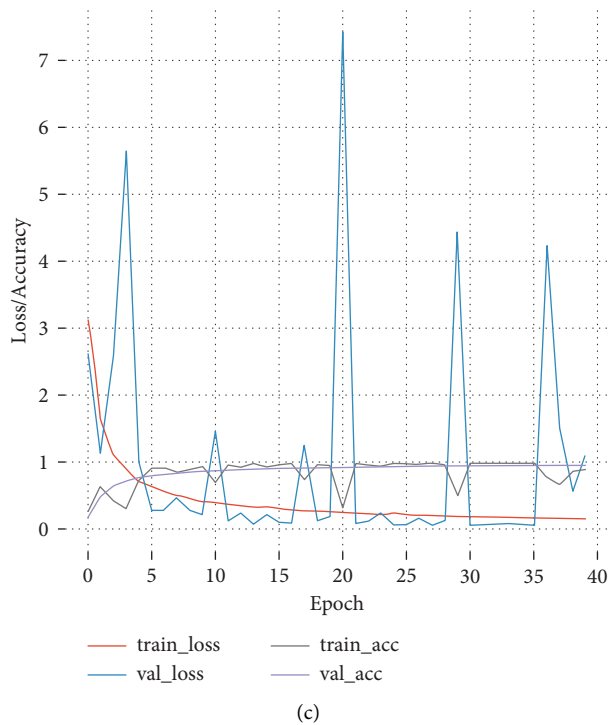
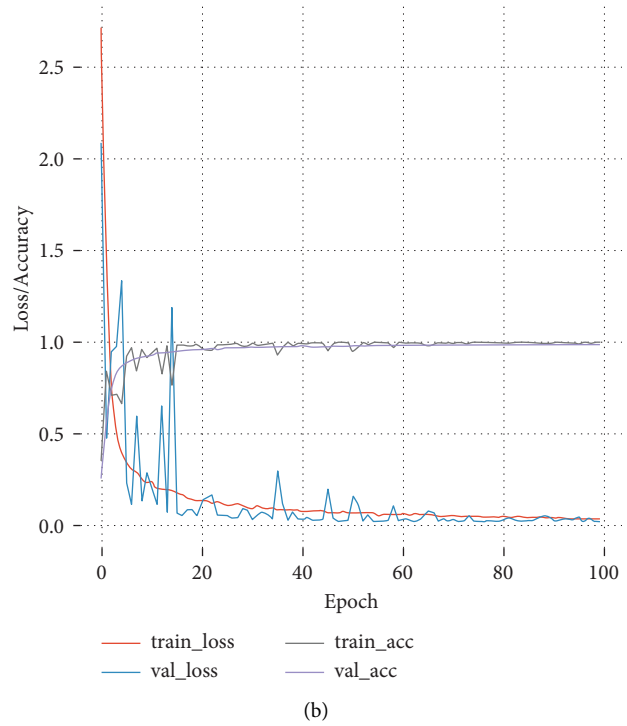
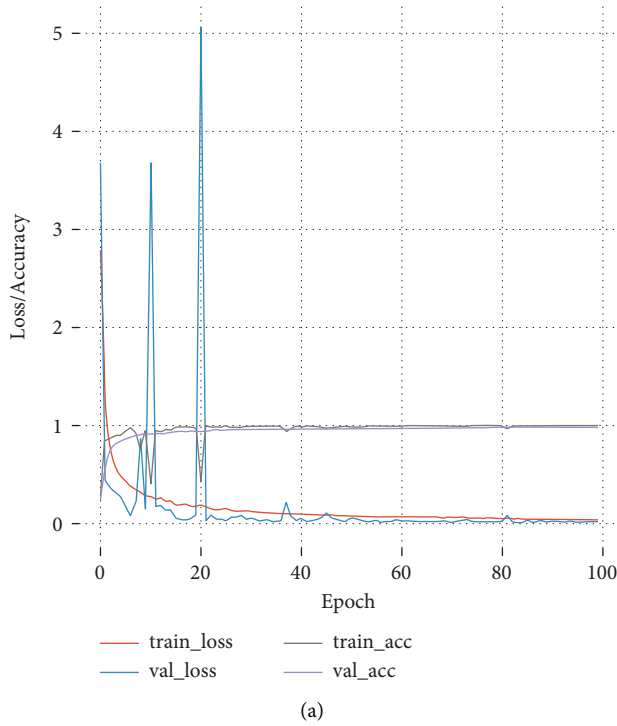


FIGURE 4: Accuracy loss graph of the proposed CNN model at (a) 100 epochs and 20 batch size, (b) 100 epochs and 40 batch size, (c) 40 epochs and 20 batch size, and (d) 40 epochs and 40 batch size.

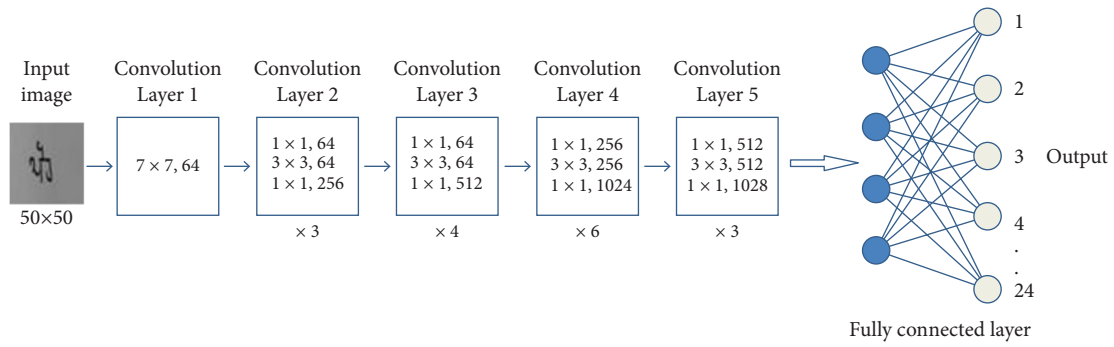


FIGURE 5: Architectural diagram of ResNet-50.

TABLE 5: Training parameters' results of ResNet50 at different epochs and batch sizes.

Results	Training parameters			
	Epoch = 100		Epoch = 40	
	Batch size 20	Batch size 40	Batch size 20	Batch size 40
Training accuracy	0.3299	0.3335	0.2503	0.2491
Validation accuracy	0.3929	0.4015	0.3031	0.3310
Training loss	2.1693	2.1658	2.4568	2.4608
Validation loss	1.9268	1.8842	2.2492	2.2037

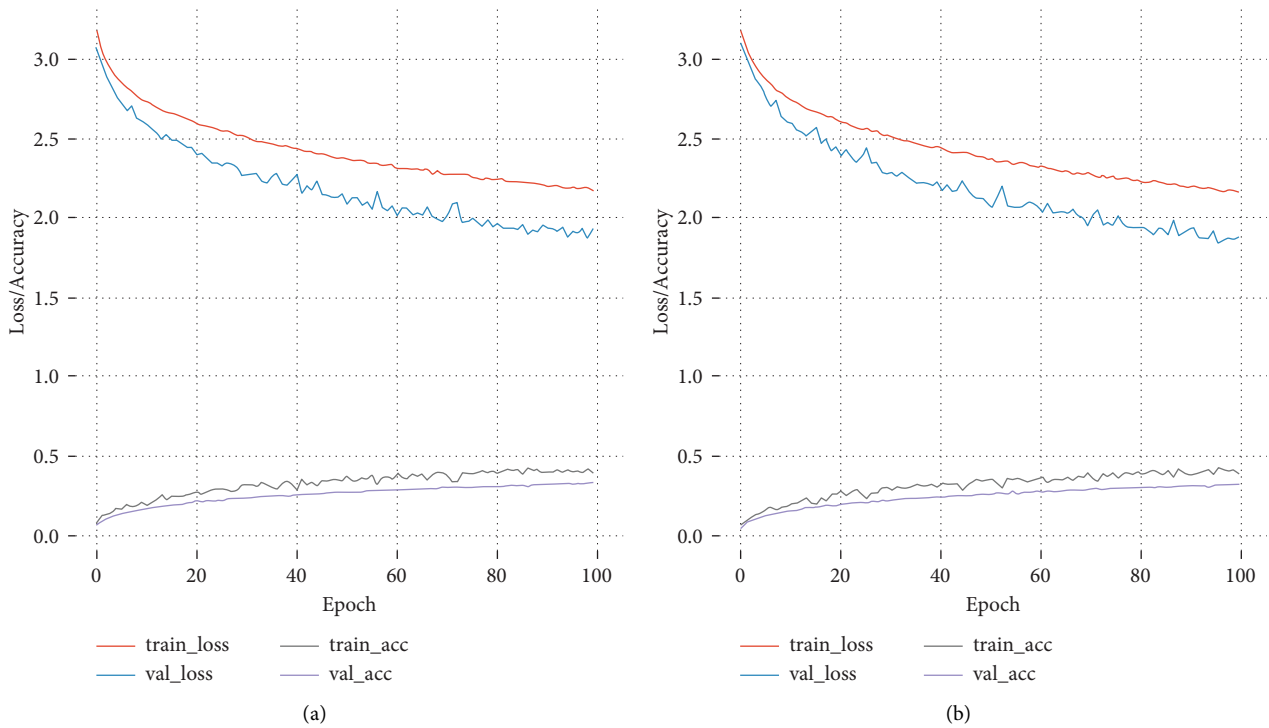


FIGURE 6: Continued.

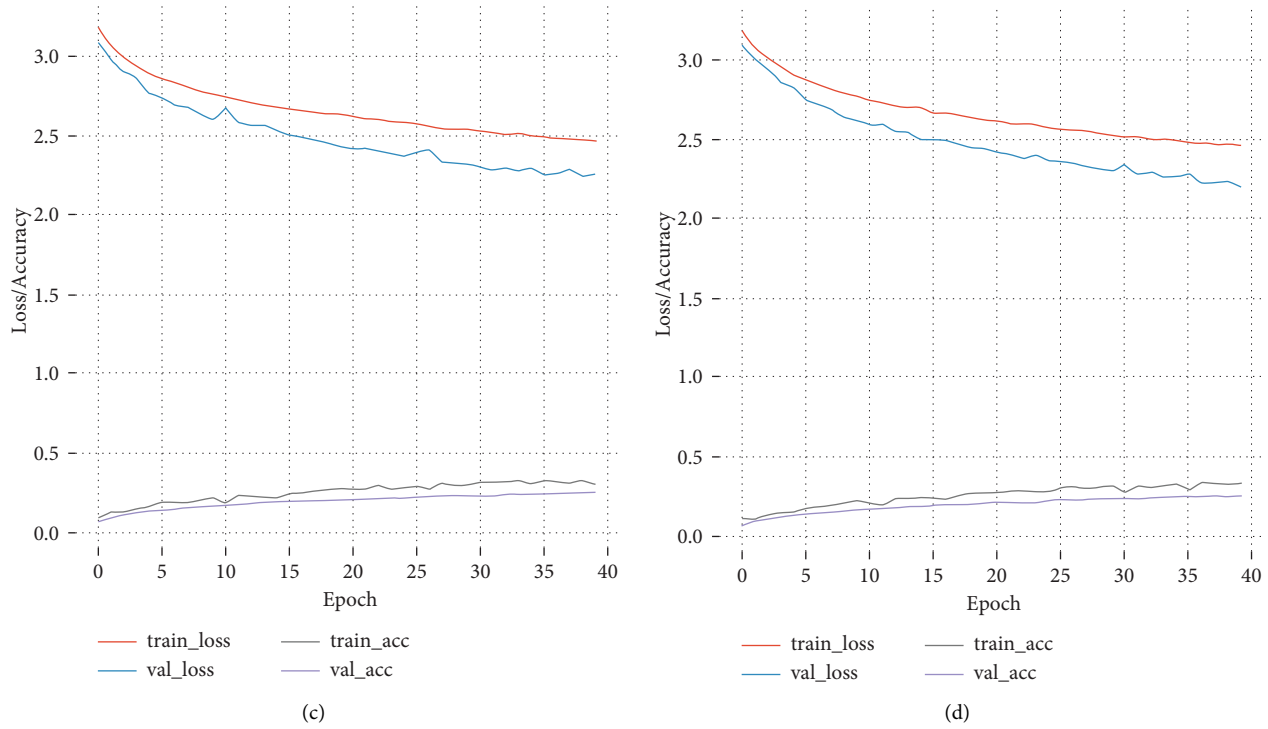


FIGURE 6: Accuracy loss graph of ResNet50 at (a) 100 epochs and 20 batch size, (b) 100 epochs and 40 batch size, (c) 40 epochs and 20 batch size, and (d) 40 epochs and 40 batch size.

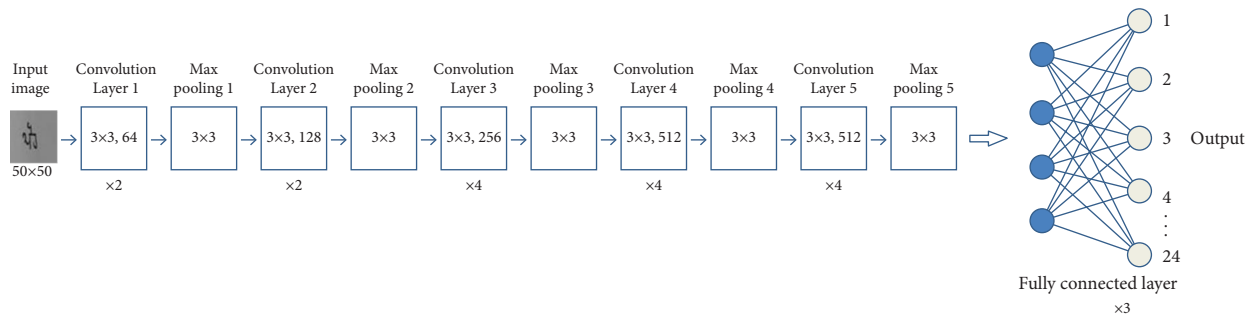


FIGURE 7: Architectural diagram of VGG19.

TABLE 6: Training parameters' results of VGG19 at different epochs and batch sizes.

Results	Training parameters			
	Epoch = 100		Epoch = 40	
	Batch size 20	Batch size 40	Batch size 20	Batch size 40
Training accuracy	0.7530	0.7452	0.6816	0.6793
Validation accuracy	0.7771	0.7758	0.7346	0.7317
Training loss	0.7560	0.7693	0.9775	0.9901
Validation loss	0.6647	0.6744	0.8217	0.8330

The accuracy loss graph for VGG16 at different epochs and batch sizes is presented in Figure 10.

It has been analyzed from the present section that the proposed CNN model as well as pretrained models including ResNet50, VGG19, and VGG16 have performed well on a self-prepared handwritten word dataset of Gurmukhi

month's name when simulated on 100 epochs and 40 batch sizes as compared to its performance in experimental settings on other training parameters such as 20 batch sizes with 100 epochs and 40 epochs with 20 and 40 batch sizes, respectively. Hence, in order to further assess the proposed CNN model's quality, a comparative analysis has been

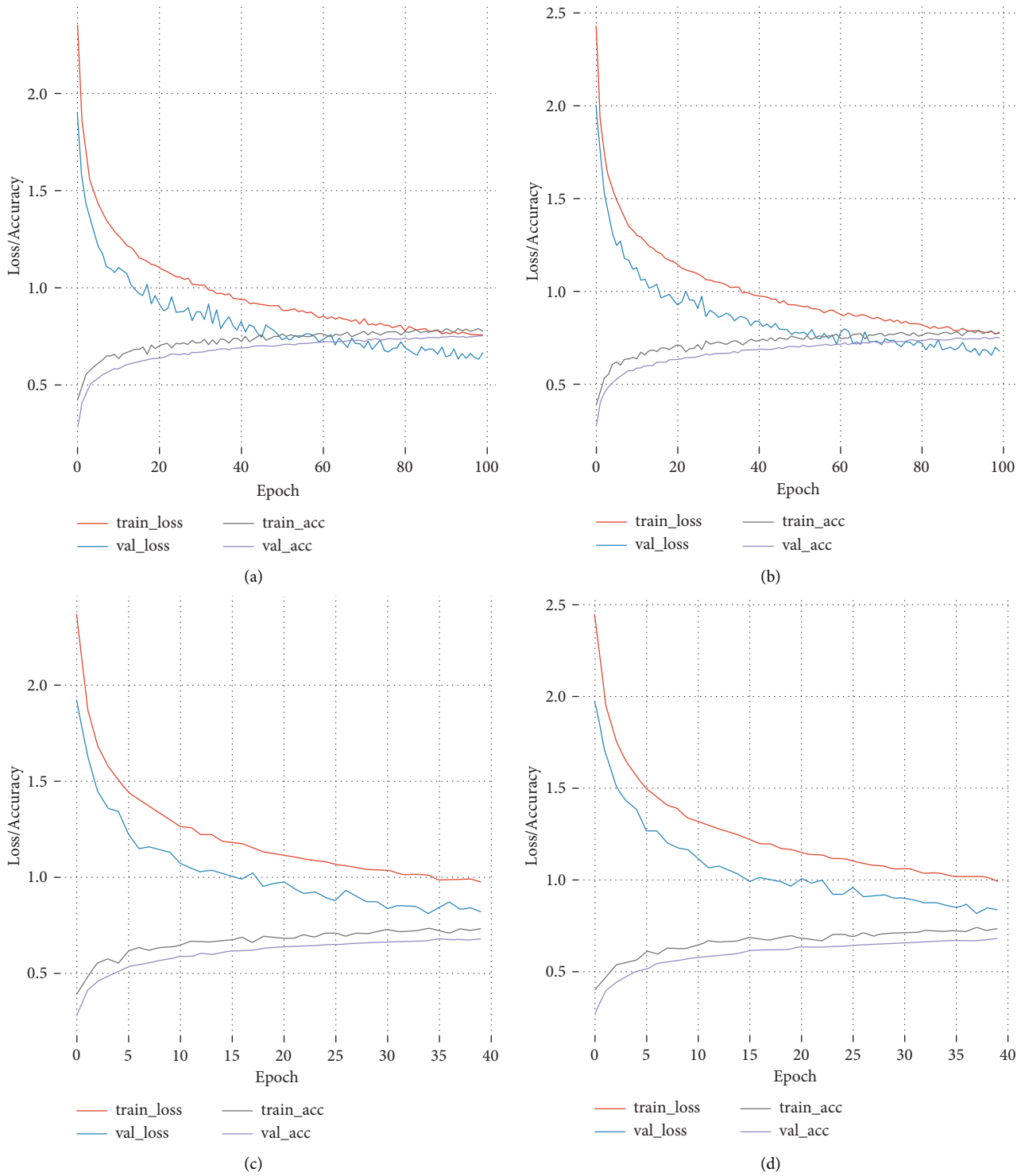


FIGURE 8: Accuracy loss graph of VGG19 at (a) 100 epochs and 20 batch size, (b) 100 epochs and 40 batch size, (c) 40 epochs and 20 batch size, and (d) 40 epochs and 40 batch size.

performed between the proposed model and a pretrained model at 100 epochs and 40 batch sizes using widely used evaluation metrics such as accuracy, F1, recall, and precision in Section 3.5.

**3.5. Comparative Analysis at 100 Epochs and 40 Batch Sizes.** It has been already motioned in the previous section that the proposed CNN model as well as pretrained models including ResNet50, VGG19, and VGG16 has been performed well on

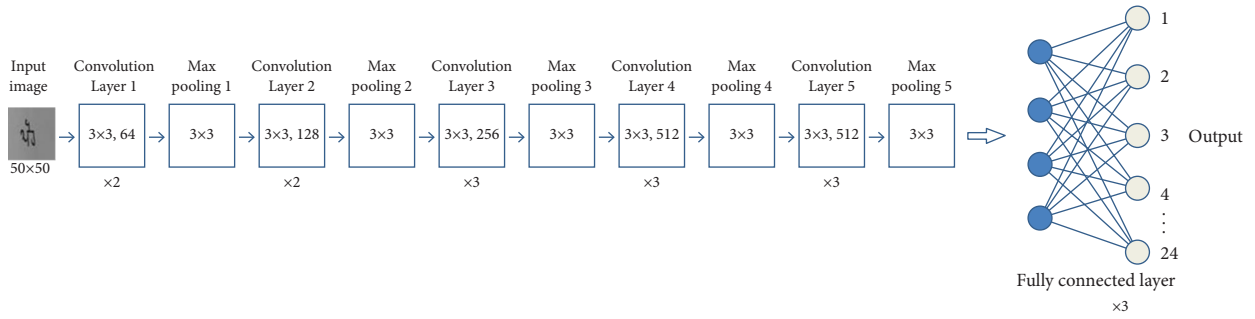


FIGURE 9: Architectural diagram of VGG16.

TABLE 7: Training parameters' results of VGG16 at different epochs and batch sizes.

Results	Training parameters			
	Epoch = 100		Epoch = 40	
	Batch size 20	Batch size 40	Batch size 20	Batch size 40
Training accuracy	0.7925	0.7832	0.7265	0.7212
Validation accuracy	0.8138	0.8056	0.7758	0.7563
Training loss	0.6274	0.6515	0.8382	0.8525
Validation loss	0.5484	0.5720	0.6908	0.7279

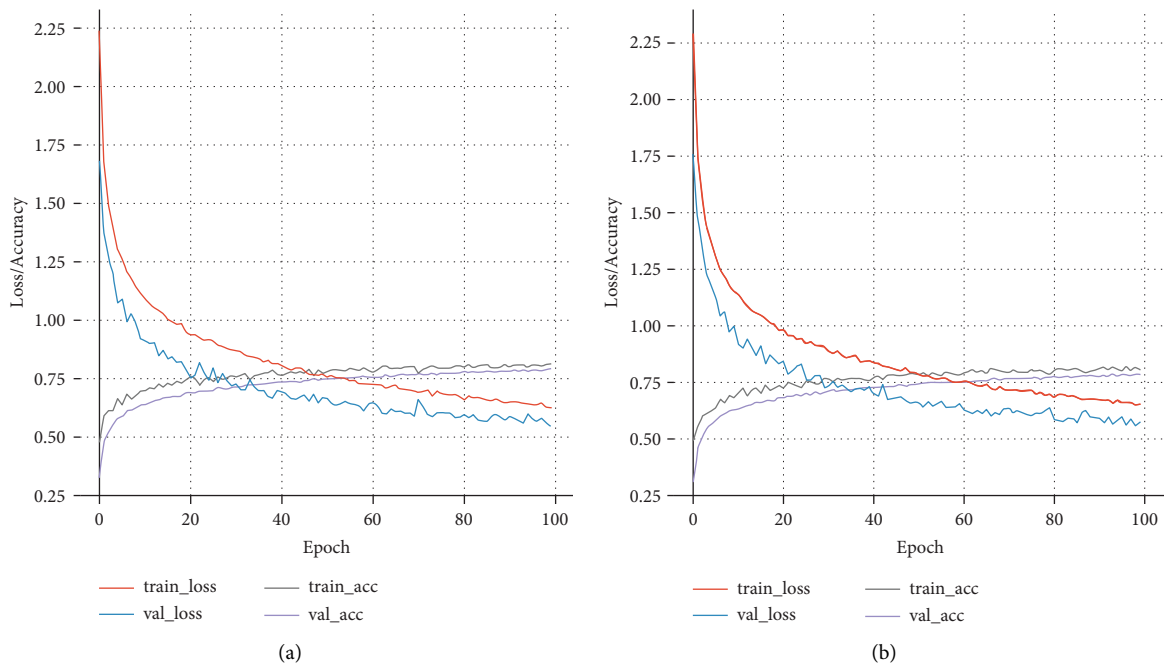


FIGURE 10: Continued.

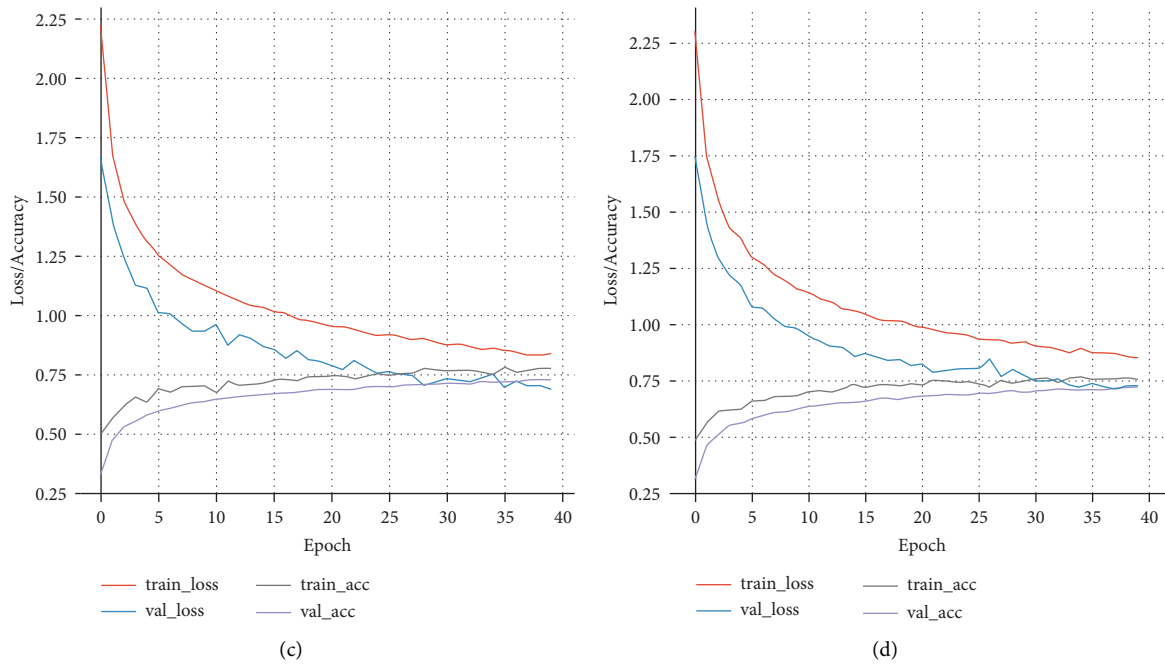


FIGURE 10: Accuracy loss graph of VGG16 at (a) 100 epochs and 20 batch size, (b) 100 epochs and 40 batch size, (c) 40 epochs and 20 batch size, and (d) 40 epochs and 40 batch size.

the self-prepared handwritten word dataset of Gurmukhi month’s name on 100 epochs and 40 batch sizes. Hence, for the comparative analysis between the models in this section, the same numbers of epochs and batch sizes have been chosen.

Furthermore, for this comparative analysis, accuracy, F1, recall, and precision have been taken as evaluation parameters.

**3.5.1. Comparative Analysis in Term of Accuracy.** The 24 class classification results of Gurmukhi month’s name dataset on the proposed model and pretrained models have been obtained using 100 epochs and 40 batch sizes. As per the obtained results, it has been noticed that the proposed CNN model has outperformed the other three pretrained models in terms of classification accuracy. The accuracy of the proposed model is 99.73%, which is almost 59.58%, 22.15%, and 19.17% more than the accuracy obtained using ResNet50, VGG19, and VGG16, respectively, as shown in Figure 11.

Other than this, Figure 12 has shown the classwise accuracy comparison of the proposed model and pretrained models.

From Figure 12, it has been observed that the proposed CNN model has given the 100% classification accuracy result for “April,” “Assu,” “June,” “Magar,” “March,” “November,” “Phagun” and “Poh” classes of Gurmukhi months, where the other three models such as ResNet50, VGG19, and VGG16 give comparatively poor accuracy on the same classes.

The least accuracy of the proposed model is on “May” and “Sawan” classes of Gurmukhi months, which is about 99.92%.

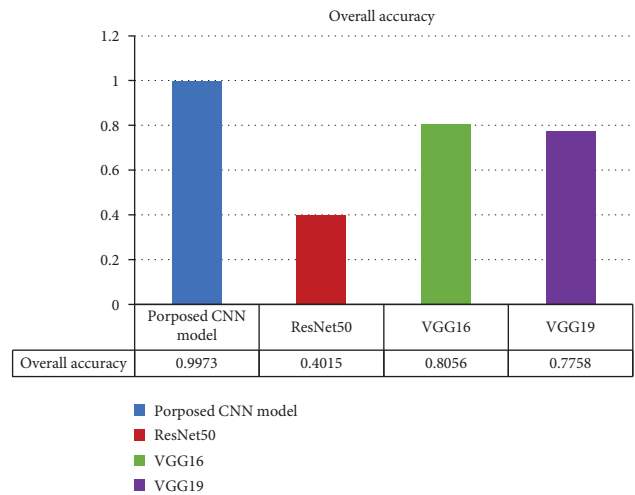


FIGURE 11: Comparison based on overall accuracy.

**3.5.2. Comparative Analysis in Term of F1 Score.** In order to test the performance of the proposed CNN model, a comparative analysis has been conducted based on the F1 score in this section. Figure 13 shows the comparison between the proposed CNN model and the other three pretrained models based on overall F1 score. It has been observed that the F1 score for the proposed CNN model has been 0.9973, which is 0.6006, 0.222, and 0.1950 higher than the F1 score value of ResNet50, VGG19, and VGG16, respectively.

In addition to this, a classwise F1 score comparison between the proposed CNN model and pretrained models is shown in Figure 14. It has been observed from Figure 14 that the proposed CNN model’s F1 score resulting in 1 for



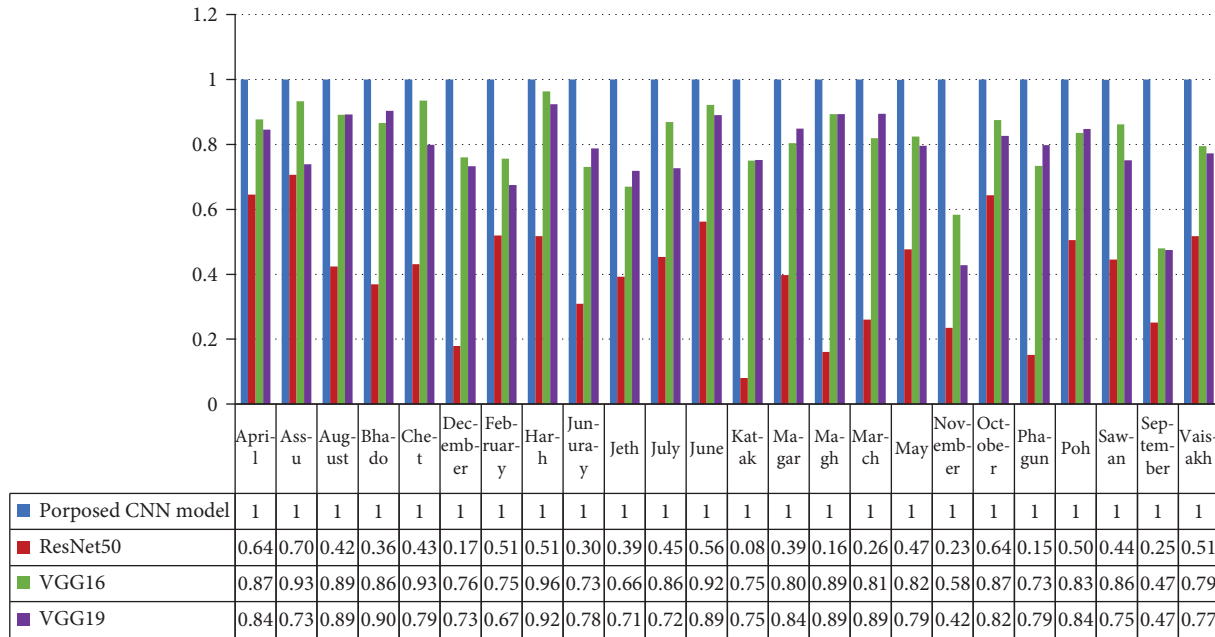


FIGURE 12: Comparison based on classwise accuracy.

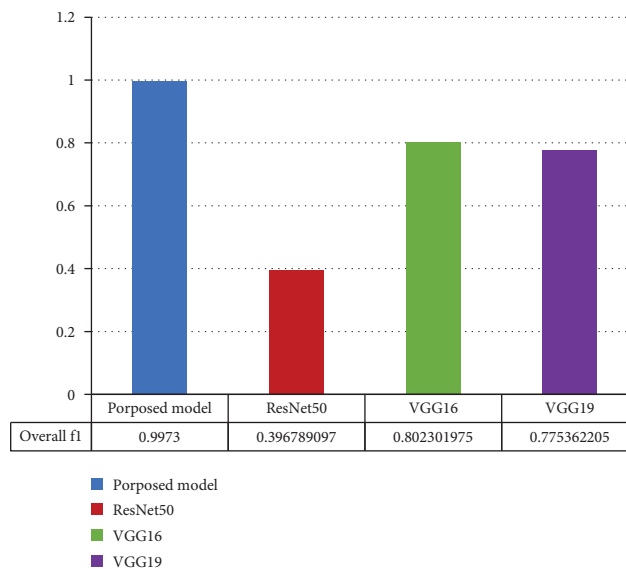


FIGURE 13: Comparison based on overall F1 score.

“April,” “Assu,” “June,” “Magar,” “March,” “November,” “Phagun,” and “Poh” month names, where the results for F1 score for the same classes from other three pretrained model have been comparatively low.

On the other hand, the least value of F1 score from the proposed CNN model has been obtained for “Vaisakh,” which is around 0.9951.

3.5.3. Comparative Analysis in Term of Recall. A comparison between the proposed CNN model and pretrained networks based on results of overall recall and classwise recall values is shown in Figures 15 and 16, respectively.

As per Figure 15, the proposed CNN model has outperformed the other three pretrained models in terms of the recall value when tested on handwritten word dataset of Gurmukhi months. The overall recall for the proposed CNN model is 0.9973 which is higher than the recall value of ResNet 50, VGG19, and VGG 16 by 0.5952, 0.2216, and 0.1931, respectively.

From Figure 16, it has been clear that the classwise recall score is resulting in 1 for “April,” “Assu,” “Bhado,” “Harh,” “January,” “June,” “Magar,” “Magh,” “March,” “May,” “November,” “Phagun,” “Poh,” “September,” and “Vaisakh” month names in case of the proposed CNN model, where F1 score’s results is comparatively poor for the same classes from other three pretrained models.

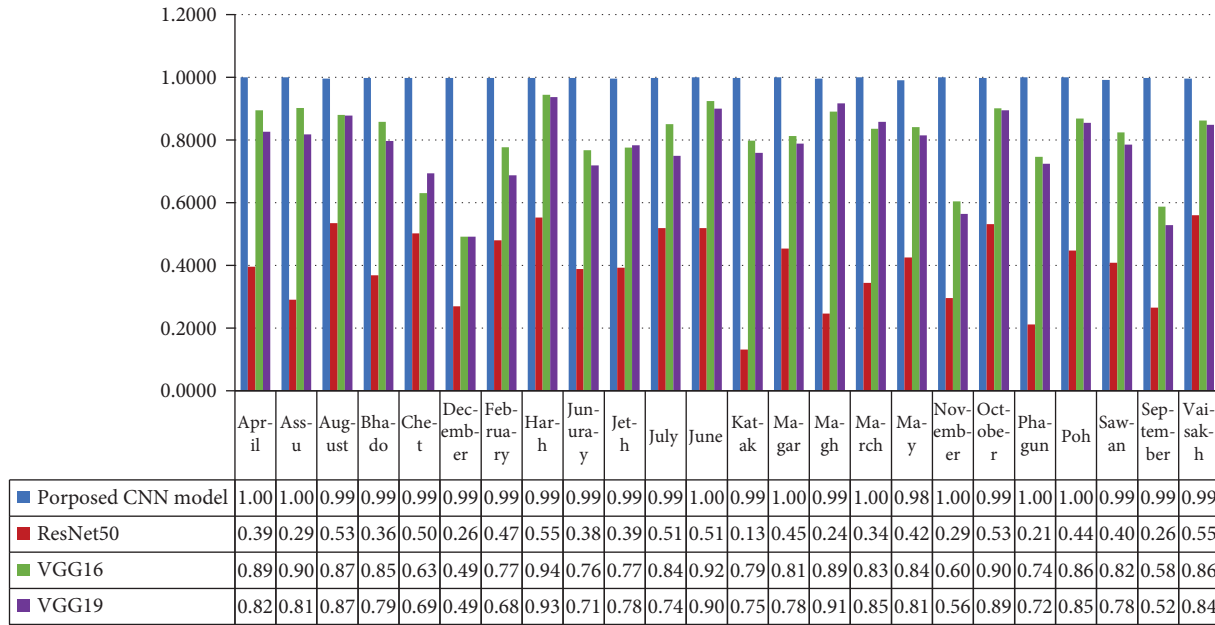


FIGURE 14: Comparison based on classwise F1 score.

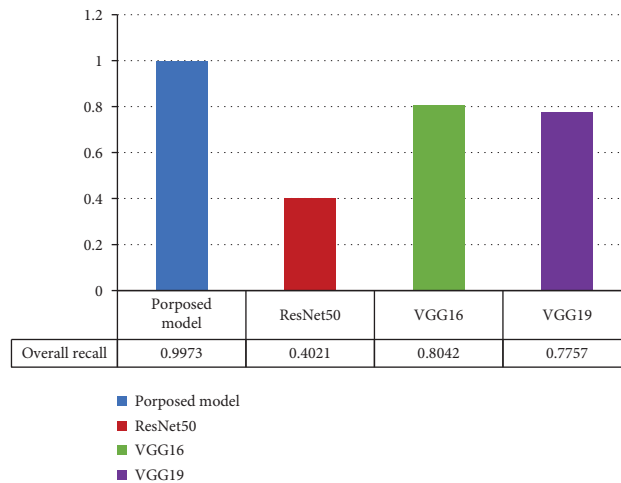


FIGURE 15: Comparison based on overall recall.

It has also been observed that the minimum value of recall is 0.9905, which is for “August” class name using the proposed CNN model.

**3.5.4. Comparative Analysis in Term of Precision.** In this section, comparative analysis in terms of the precision value has been performed for 24-class classification results of Gurmukhi month’s name dataset using the proposed model and pretrained models at 100 epochs and 40 batch sizes. The comparative analysis in Figure 17 depicted that the proposed CNN model has outperformed the other three pretrained models in terms of precision. As the results of precision using the proposed CNN model is 0.9974, which is almost 0.5446, 0.2026, and 0.1801 more than the precision value obtained using ResNet50, VGG19, and VGG16, respectively, as shown in Figure 17.

A classwise precision comparison between the proposed CNN model and pretrained models, ResNet 50, VGG19, and VGG16, is shown in Figure 18. It has been observed from Figure 18 that, for class name “April,” “Assu,” “August,” “Chet,” “December,” “February,” “Jeth,” “July,” “June,” “Katak,” “Magar,” “March,” “November,” “October,” “Phagun,” and “Poh,” the obtained value of precision is one using the proposed CNN model. It has also been observed that, by using other three pretrained models, the precision value has been comparatively low for these classes.

Using the proposed CNN model, the least precision value (0.9903) has been obtained for class name “Vaisakh.”

From the comparative analysis performed in this section, it has been found that the proposed CNN model has outperformed the other three pretrained models in terms of four performance assessment matrices such as accuracy, recall, precision, and F1 score when simulated on

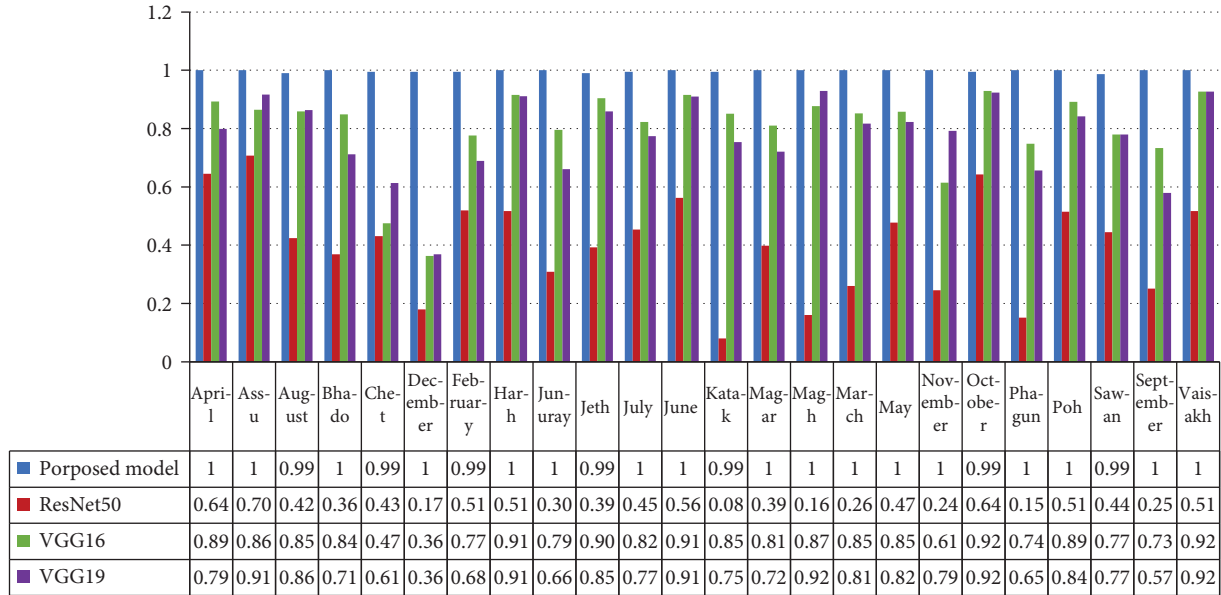


FIGURE 16: Comparison based on classwise recall.

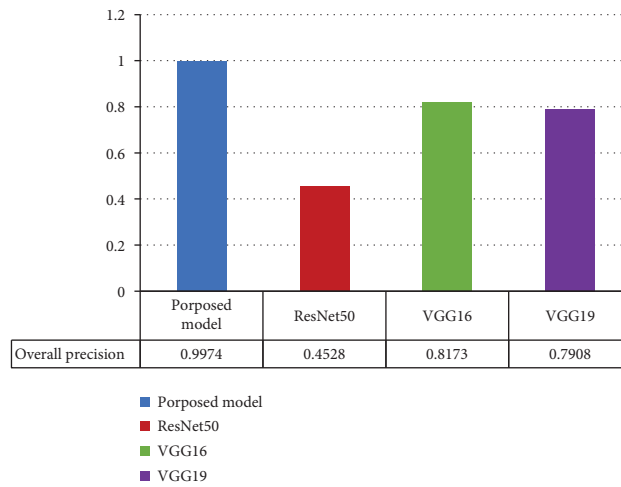


FIGURE 17: Comparison based on overall precision.

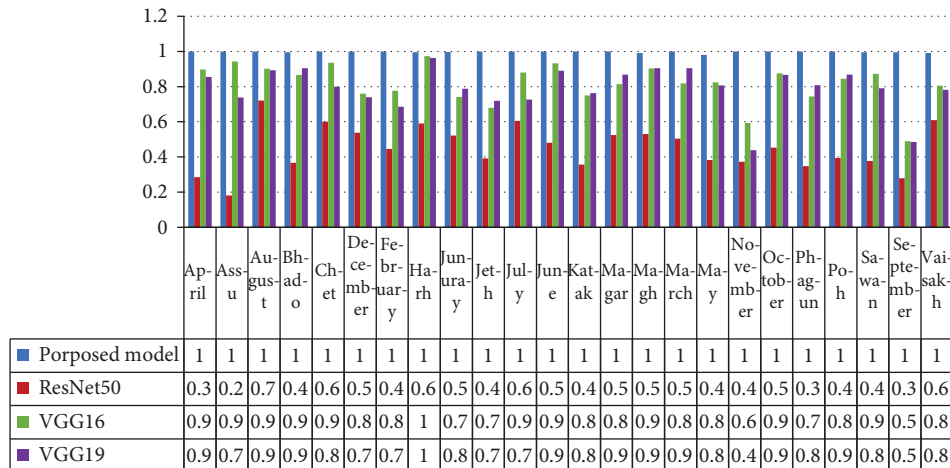


FIGURE 18: Comparison based on classwise precision.

a self-prepared handwritten word dataset of Gurmukhi month's name for image classification. Hence, the proposed CNN model is the best fit model for various regional applications of month's name classification.

#### 4. Conclusion

In this research article, the classification of a handwritten word dataset of Gurmukhi month's name has been done into 24 different classes. For this, the CNN model has been prepared from scratch with five convolutional, three pooling layers, one flattening layer, and one dense layer. Furthermore, a performance assessment of the proposed CNN model has been conducted at different numbers of epochs and batch sizes, and it has been observed that the proposed CNN model has performed well at epoch 100 and 40 batch sizes. In addition to this, a comparative analysis has also been performed between the proposed CNN model and three pretrained models named as ResNet50, VGG19, and VGG16 based on different performance assessment matrixes such as accuracy, F1 score, recall, and precision. From this experimentation, it has been concluded that the proposed CNN model at 100 epochs and 40 batch sizes outperformed the other three pretrained models in the classification of handwritten words dataset in 24 classes.

#### Data Availability

The data used to support the findings of the study are available from the corresponding author upon request.

#### Conflicts of Interest

The authors declare that they have no conflicts of interest.

#### References

- [1] A. Sharma, R. Kumar, and R. K. Sharma, "Online handwritten gurmukhi character recognition using elastic matching," *2008 Congress on Image and Signal Processing*, vol. 2, pp. 391–396, 2008.
- [2] A. Sharma, R. Kumar, and R. K. Sharma, "Rearrangement of recognized strokes in online handwritten gurmukhi words recognition," in *Proceedings of the 2009 10th International Conference on Document Analysis and Recognition*, Barcelona, Spain, July 2009.
- [3] R. Dhir, "Moment based invariant feature extraction techniques for bilingual character recognition," in *Proceedings of the 2010 2nd International Conference on Education Technology and Computer*, Shanghai, China, June 2010.
- [4] M. Kumar, M. K. Jindal, and R. K. Sharma, "Classification of characters and grading writers in offline handwritten Gurmukhi script," in *Proceedings of the 2011 International Conference on Image Information Processing*, Shimla, India, September 2011.
- [5] M. Kumar, R. K. Sharma, and M. K. Jindal, "Efficient feature extraction techniques for offline handwritten gurmukhi character recognition," *National Academy Science Letters*, vol. 37, no. 4, pp. 381–391, 2014.
- [6] R. Kumar, R. K. Sharma, and A. Sharma, "Recognition of multi-stroke based online handwritten gurmukhi aksharas," *Proceedings of the National Academy of Sciences, India, Section A: Physical Sciences*, vol. 85, no. 1, pp. 159–168, 2015.
- [7] K. Verma and R. K. Sharma, "Recognition of online handwritten Gurmukhi characters based on zone and stroke identification," *Sadhana*, vol. 42, no. 5, pp. 701–712, 2017.
- [8] M. Kumar, M. K. Jindal, and R. K. Sharma, "Offline handwritten gurmukhi character recognition: analytical study of different transformations," *Proceedings of the National Academy of Sciences, India, Section A: Physical Sciences*, vol. 87, no. 1, pp. 137–143, 2017.
- [9] H. Singh, R. K. Sharma, and V. P. Singh, "Efficient zone identification approach for the recognition of online handwritten Gurmukhi script," *Neural Computing & Applications*, vol. 31, no. 8, pp. 3957–3968, 2019.
- [10] M. Kumar, M. K. Jindal, R. K. Sharma, and S. R. Jindal, "A novel framework for writer identification based on pre-segmented Gurmukhi characters," *Sadhana*, vol. 43, no. 12, 2018.
- [11] Sakshi, N. K. Garg, and M. Kumar, "Writer identification system for handwritten gurmukhi characters: study of different feature-classifier combinations," in *Proceedings of International Conference on Computational Intelligence and Data Engineering*, Springer, Singapore, 2018.
- [12] M. Kumar, S. R. Jindal, M. K. Jindal, and G. S. Lehal, "Improved recognition results of medieval handwritten gurmukhi manuscripts using boosting and bagging methodologies," *Neural Processing Letters*, vol. 50, no. 1, pp. 43–56, 2019.
- [13] A. Garg, M. K. Jindal, and A. Singh, "Degraded offline handwritten Gurmukhi character recognition: study of various features and classifiers," *International Journal of Information Technology*, vol. 14, no. 1, pp. 145–153, 2019.
- [14] M. Kumar, M. K. Jindal, R. K. Sharma, and S. R. Jindal, "Performance evaluation of classifiers for the recognition of offline handwritten Gurmukhi characters and numerals: a study," *Artificial Intelligence Review*, vol. 53, no. 3, pp. 2075–2097, 2020.
- [15] T. Ganji, M. S. Velpuru, and R. Dugyala, "Multi variant handwritten Telugu character recognition using transfer learning," *IOP Conference Series: Materials Science and Engineering*, vol. 1042, no. 1, Article ID 012026, 2021.
- [16] R. Pramanik and S. Bag, "Handwritten Bangla city name word recognition using CNN-based transfer learning and FCN," *Neural Computing & Applications*, vol. 33, no. 15, pp. 9329–9341, 2021.
- [17] R. Pramanik, P. Dansena, and S. Bag, "A study on the effect of CNN-based transfer learning on handwritten indic and mixed numeral recognition," in *Document Analysis and Recognition*, Springer, Singapore, 2019.
- [18] U. Jindal, S. Gupta, V. Jain, and M. Paprzycki, "Offline handwritten Gurumukhi character recognition system using deep learning," in *Advances in Bioinformatics, Multimedia, and Electronics Circuits and Signals*, Springer, Singapore, 2020.
- [19] R. Geetha, T. Thilagam, and T. Padmavathy, "Effective offline handwritten text recognition model based on a sequence-to-sequence approach with CNN–RNN networks," *Neural Computing & Applications*, vol. 33, no. 17, Article ID 10923, 2021.
- [20] S. Singh, A. Sharma, and V. K. Chauhan, "Online handwritten Gurmukhi word recognition using fine-tuned Deep Convolutional Neural Network on offline features," *Machine Learning with Applications*, vol. 5, Article ID 100037, 2021.
- [21] M. Ragab, A. S. A.-M. Al-Ghamdi, B. Fakhieh, H. Choudhry, R. F. Mansour, and D. Koundal, "Prediction of diabetes

- through retinal images using deep neural network,” *Computational Intelligence and Neuroscience*, vol. 2022, Article ID 7887908, 6 pages, 2022.
- [22] S. Shambhu, D. Koundal, P. Das, and C. Sharma, “Binary classification of COVID-19 CT images using CNN: COVID diagnosis using CT,” *International Journal of E-Health and Medical Communications*, vol. 13, no. 2, pp. 1–13, 2021.
- [23] R. Nair, A. Alhudhaif, D. Koundal, R. I. Doewes, and P. Sharma, “Deep learning-based COVID-19 detection system using pulmonary CT scans,” *Turkish Journal of Electrical Engineering and Computer Sciences*, vol. 29, no. 1, pp. 2716–2727, 2021.
- [24] D. J. Jwo, S. F. Chiu, S. Gupta et al., “Deep learning based automated detection of diseases from apple leaf images,” *Computers, Materials & Continua*, vol. 71, no. 1, pp. 1849–1866, 2022.