Deep Convolutional Neural Network Approach for Classification of Poems

Rushali Deshmukh[1(✉)] and Arvind W. Kiwelekar[2]

[1] Department of Computer Engineering, JSPM's Rajarshi Shahu College of Engineering, Pune 411033, India
radeshmukh_comp@jspmrscoe.edu.in

[2] Department of Computer Engineering, Dr. Babasaheb Ambedkar Technological University, Lonere 402103, Raigad (M. S.), India
awk@dbatu.ac.in

Abstract. In this paper, we proposed an automatic convolutional neural network (CNN)-based method to classify poems written in Marathi, one of the popular Indian languages. Using this classification, a person unaware of Marathi Language can come to know what kind of emotion the given poem indicates. To the best of our knowledge, this is probably the first attempt of deep learning strategy in the field of Marathi poem classification. We conducted experiments with different models of CNN, considering different batch sizes, filter sizes, regularization methods like dropout, early stopping. Experimental results witness that our proposed approach outperforms both in effectiveness and efficiency. Our proposed CNN architecture for the classification of poems produces an impressive accuracy of 73%.

Keywords: Classification · Convolution · Convolutional neural network · Poem classification

1 Classification

Classification is the task of assigning one of the predefined classes to given data. It can be binary or multiclass classification. We aim to build a model for the automatic classification of poems. Traditional machine learning algorithms don't perform well with massive data. Gu et al. [1] surveyed recent advances in convolutional neural networks (CNN) and introduced applications of CNN in speech, computer vision, and natural language processing. In our approach, we considered around 3000 poems of 9 classes. We explored various CNN models considering the number of filters, their sizes, and batch size as a hyperparameter for the classification of poems. To avoid overfitting, 'dropout' and 'early stopping' regularization techniques are used. First, we did a survey of classification using deep learning. We also surveyed classification techniques for the Marathi language. The rest of the paper is organized as follows, Sect. 1 contains the introduction

J.-H. Kim et al. (Eds.): IHCI 2021, LNCS 13184, pp. 74–88, 2022.
https://doi.org/10.1007/978-3-030-98404-5_7

of classification, Sect. 2 contains the related work of classification using CNN, Sect. 3 presents different models using CNN for Marathi poem classification, Sect. 4 concludes with the conclusion and future scope.

2 Literature Survey

Kowsari et al. [2] surveyed various classification algorithms such as Naive Bayes, Decision tree, Random forest, Support vector machines, K nearest neighbor, Deep learning etc. and reported advantages and limitations of each of it. Minaee et al. [3] discussed more than 150 deep learning models like feed forward neural network, RNN-Based models, CNN-Based models, Capsule neural networks, Models with attention mechanism, Memory augmented networks, Graph neural networks, Siamese neural networks, Hybrid models, Transformers and Pretrained language models and their strengths for classification of text. Kamath et al. [4] classified Health Dataset from an insurance company with 13500 documents into 18 classes and Tobacco-3482 dataset of images related to tobacco with 2800 documents into nine classes. Maximum accuracy of 73% was achieved on raw data and 67% on processed data for the tobacco dataset among all traditional machine learning algorithms for Health data. For Tobacco data, maximum accuracy of 81% and 77% accuracy was achieved on raw and processed data respectively using machine learning algorithms. CNN gave 82% accuracy on raw data and 96% on processed Health data. For Tobacco data, 84% and 89% accuracy was achieved on processed and raw data respectively. Georgakopoulos et al. [5] classified Toxic Comment using three convolutional layers with a filter size width of 128 and a height of 3,4 and 5 respectively. After each convolutional layer, the max-pooling layer was used. The last pooling layer's output concatenates to a fully connected layer, which was connected to the output layer, which uses a softmax function. The accuracy obtained was around 91%, while for traditional machine learning algorithms, it was in the range of 65% to 85%. Cano and M. Morisio [6] implemented Ngram CNN architecture with the following layers, 1) Embedding layer 2) Convolution layer with a filter of 1-gram, 2-gram, ..., Ngram with Relu activation function 3) Max pooling layer/4, which pools max value from four values 4) Convolution layer with a filter of 1-gram, 2-gram, ..., Ngram with Relu activation function 5) Max pooling layer/4 6) Dense layer of 100 units and L2 regularization with 0.09 weight 7) Dropout layer with 0.5 dropout rate 8) Output layer. MLPN dataset 2,500 positive and 2,500 negative song lyrics, and IMDB Movie review dataset of 50K size with two classes positive and negative and Mobile Phone reviews of size 232 k with positive and negative classes used for experimentation. 1) For the MLPN dataset, the accuracy achieved was 75.6%. 2) IMDB Movie review dataset accuracy achieved was 91.2%. 3) Mobile Phone review accuracy achieved was 95.9%. Hughes [7] classified Medical documents each of 50 words with a vector of size 100. The model consists of the following layers. 1) Embedding layer 2) Convolution layer of filter size 5 and 256 filters ReLU activation 3) Max pooling 4)Convolution layer of filter size 5 and 256 filters and ReLU activation 5) Max pooling 6) Dropout layer with a rate 0.5 7) Fully connected layer of size 128 and ReLU 8) Dropout layer with rate 0.5 9) Output layer of size 26 with softmax activation. Twenty-six categories were used with 4000 sentences of each category of clinical text. For validation, 1000 sentences from each category were used. The accuracy achieved was 0.68%. Hsu and Tzuhan [8] did Petroleum Engineering Data Text Classification Using Convolutional Neural Network Based Classifier. They used the Raw dataset of 400,000 texts with six classes. The accuracy achieved with CNN was 85.83%. Kalchbrenner et al. [9] did Binary and Multiclass classification of Twitter sentiment prediction and TREC question dataset. Dynamic Convolutional Neural Network was implemented for a seven-word input sentence with word embedding of 4. Following layers were used: 1) Convolutional layer with two feature maps and width of filter was 3, 2) Dynamic k max-pooling with k = 5, 3) Convolutional layer with two feature maps and width of filter was 2, 4) folding, 5) Dynamic k max-pooling with k = 3, 6) Fully connected layer. For Movie reviews dataset with binary and multiclass classification accuracy obtained was in the range of 86.8%, for Twitter sentiment accuracy achieved was 87.4%, and for Question classification accuracy was 93%. Kim [10] implemented the CNN-rand model in which all words are randomly initialized, CNN-static model with pre-trained vectors from word2vec, CNN-nonstatic with fine-tuned pre-trained vectors and CNN-multichannel using two sets of word vectors. For MR dataset among existing models using CNN-nonstatic highest accuracy of 81.5% was obtained. For SST-2 dataset accuracy of 88.1%, for CR dataset 85.0% of accuracy was obtained using CNN-multichannel. For the MPQA dataset using CNN-static 89.6% accuracy was achieved. For sentence classification, Zhang and Wallace [11] fine-tuned parameters like the effect of filter region size, number of feature maps for each filter region size, number of feature maps for each filter region size, multiple region size, activation functions, pooling strategy, the effect of regularization. For datasets MR, SST-1, SST-2, Subj, TREC, CR, MPQA, Opi, Irony, they provided guidelines for finding the best configuration of a model for a given dataset. De Sousa Pereira Amorim [12] implemented Logistic Regression and Deep Learning for 13,651 tweets in Portuguese with 6851 of the political class and 6800 of non-political class. With Logistic Regression accuracy achieved Severyn and Moschitti implemented message and phrase was 90.5%, and Deep learning accuracy obtained was 83.8%. Severyn and Moschitti [13] implemented message and phrase-level twitter sentiment classification using three approaches 1) Randomly initialized parameters 2) an unsupervised neural language model to initialize word embeddings, which are further tuned. Baker et al. [14] used Convolutional Neural Networks for Biomedical text classification of cancer domain dataset. After the embedding layer, they used convolutions of different filter sizes and 1-max pooling and finally fully connected layer. They achieved an F-score of 81.0%.

An earlier study for classification of Indian languages like Urdu, Punjabi, and Tamil was mainly focused using machine learning techniques like Naive Bayes, SVM, Artificial neural network [15–17]. For Marathi language, fourth spoken language in India Label Induction Grouping (LINGO) algorithm was used to categirize 200 documents into 20 categories by patil and Bogiri [18]. For 1170

Marathi poems of 5 categories, Support vector machine was used in the approach of Deshmukh et al. [19].

The above literature survey shows that text classification for Indian languages is mainly focused on using traditional machine learning algorithms. Classification using attention CNN+LSTM deep learning for 9142 poems in the English language was done in the approach of Ahmed et al. [20] and acheived accuracy of 88%. Our approach is focused on the classification of around 3000 poems of 9 categories using a convolutional neural network.

3 Classification of Poems Using Deep Learning

Here we gathered Marathi poems from https://marathikavita.co.in/. Details are as shown in Table 1. 80% of the dataset is used for training, and 20% is used for testing. Out of 2803 poems, 561 poems are used for testing, and 2242 poems are used for training. A stratified train/test split is used. It makes split so that the same proportion of values from each class of dataset appears in train and test data.

Table 1. Category wise poem count.

Sr. no.	Type of poem	No. of poems
1	Badbad	207
2	Bhakti	465
3	Gambhir	330
4	Maitri	211
5	Motivation	286
6	Prem	372
7	Shrungar	316
8	Vidamban	345
9	Vinodi	271
Total		**2803**

From training data, 20% is used for validation purpose. We experimented with CNN models with the following approaches.

- Effect of region size,
- Effect of multiple regions,
- Effect of batch size.

3.1 Convolutional Neural Network Approach

In earlier days, CNN's were mainly used for Image processing. Nowadays, it has been successful in text classification tasks. There are three types of layers in CNN namely convolutional layers, pooling layers, and fully connected layers [21]. Convolution operation reduces the number of free parameters. The pooling layer also reduces the number of parameters by reducing the spatial size of the representation. It operates on each feature map independently. The common approach used is max pooling. Max pooling partitions input into nonoverlapping regions and from each region partition outputs maximum. Another approach is average pooling or L2-norm pooling outputs average from each region. After several convolutional and pooling layers at the end of CNN, a fully connected layer of neurons is used that have full connections to all activation's in the previous layer. Figure 1 shows that the sentence of size seven words is represented using a vector of the dimension of 5. Convolution is done using the region sizes of 2, 3, 4 with six filters, 2 for each region.

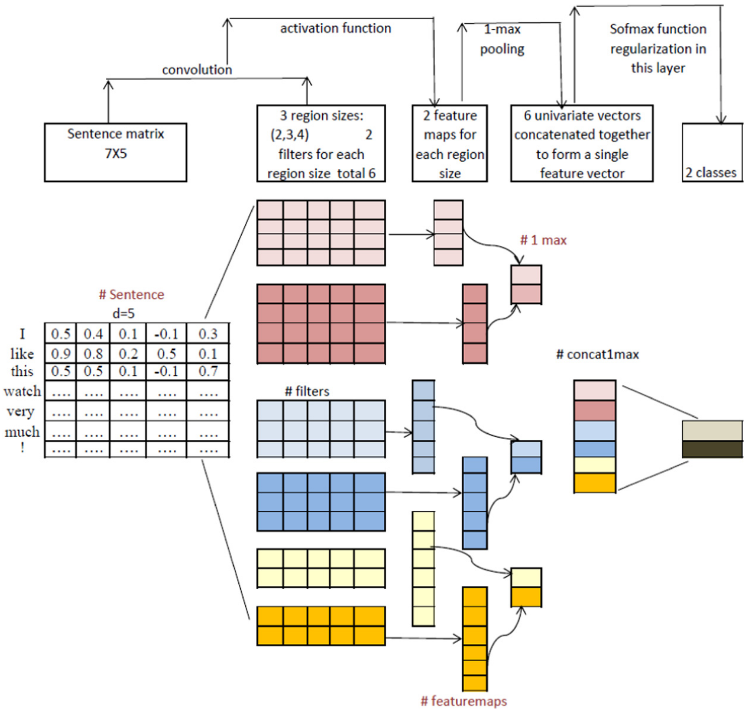

Fig. 1. CNN for classification example.

Let us consider how CNN performs convolutions. Some numbers are filled out for clarity in the matrix of the sentence and the filter matrix. Figure 2 shows the action of a 2-word filter on the sentence matrix using convolution as a matrix product [22]. A filter is of size 2*5, where 2 is region size. First, it performs convolution on the first two words 'I' and 'Like'.

It performs an element-wise product with a filter, then sums to obtain the final value.

(0.6*0.2+0.5*0.1+0.2*0.2..........+0.1*0.1 = 0.51).

The first value in the output is computed. The next filter moves one word down and overlays with 'like' and 'this' words. Output O has a shape of (S−R+1*1), where S is sentence size, and R is region size. Next, we add bias term to it, then activation function ReLU is applied to obtain a feature map of the same size (S−R+1*1) after that 1-max pooling is applied to obtain max value from each feature map. After that, obtained six univariate vectors are concatenated to obtain a single feature vector. Finally, the softmax activation function in the output layer classifies input into two classes.

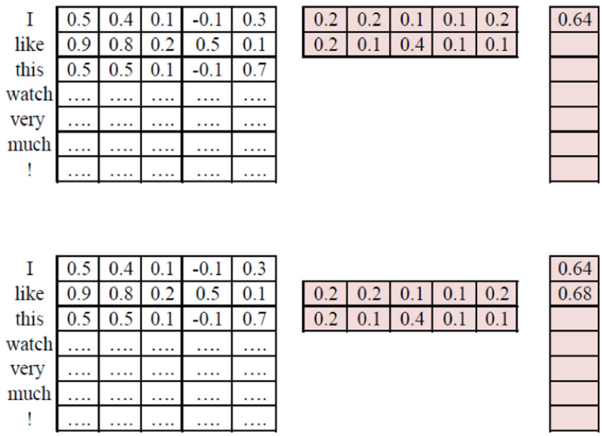

Fig. 2. Convolution example.

In our case, vocabulary size is 2000. Maximum words for each document is 200. If a document word size is less than 200, post padding is done. Using the embedding layer of Keras for each word, a word vector of size 100 is generated.

The output of the embedding layer is given to the convolution layer for filtering of different region sizes of 2, 3, 4, 6, 8, 10. After filtering, feature maps are generated using activation function ReLU. The next layer is a dropout layer with a 0.5 dropout value. Then Max-pooling layer pools 50% max values. The next flatten layer generates a 1-D vector, which is given as input to a dense layer of 256 size with the ReLU activation function. The final output layer generates probabilities for nine poem classes using a softmax function. As it is a multiclass classification, 'categorical_crossentropy' is the loss function used, and 'adam' is the optimizer used. Adam optimizer is computationally efficient with little memory requirements and suitable for problems of large data and/or parameters [23].

Dropout is used to avoid the overfitting of neural networks. During training, dropout regularization randomly drops units from the neural network [24].

For regularization, 'early stopping' is used, which monitors the value of loss for test data and immediately stops training of model if performance on test data becomes worst. The batch size is 32 with ten epochs or iterations. We have also explored the effect of different batch sizes, the effect of region sizes, the effect of multichannel with different region sizes.

Table 2. Effect of filter region size.

Region size	Accuracy on validation data	Accuracy on test data
1	0.66	0.65
2	0.68	0.65
4	0.64	0.68
6	0.61	0.66
8	**0.69**	**0.71**
10	0.64	0.65

Effect of Filter Region Size: We considered the effect of region sizes of 1, 2, 4, 6, 8 and recorded the accuracy of validation and test data as shown in Table 2. For eight region size with 100 filter model built is as shown in Fig. 3. For each document of size 200*100, eight size filter of size 8*100 performs filtering of size 8*100. Each filter is initialized to random weights. It performs an element-wise product with a filter. Then filter moves down by one word. The output of it has a shape of $(200-8+1*1) = 193$. Feature map is generated by adding bias, then activation function ReLU. One hundred feature maps of size 193 are generated for 100 filters. After a convolution dropout layer with a 0.5 dropout value is used. Then max-pooling layer pools 50% max values from each feature map. Output after Maxpooling is 100 feature maps with 96 values. Next Flatten layer generates a 1-D vector of size $96*100 = 9600$ values, which is connected to a dense layer of 256 neurons. The final output layer applies a softmax function and generates probabilities for nine classes. Figure 4 shows that as no. of epochs increases loss decreases for training data, but for validation data after six epochs, loss increases. As we have used early stopping, it will stop the training of the model after seven epochs. Figure 5 shows epochwise accuracy for train and validation data.

Classwise precision, recall, f1-score is shown in Table 3.

The average precision score, micro-averaged over all classes for model of Fig. 3 is 0.72. Table 3 shows that the maximum F1-score of 0.83 is obtained for 'Bhakti' type of poems. Confusion matrix in Fig. 6 shows that 13 poems from the 'gambhir' category are classified in the 'vidamban' category. For 'motivation' class 11 poems are classified in the 'vidamban' class. Because of that, precision for 'vidamban' is 0.48. Recall for motivation is 0.54. Figure 7 shows class wise ROC curves. The area for ROC curves is in the range of 0.80 to 0.89, except for the 'motivation' class is 0.75. The micro-average of the ROC curve is 0.84, and the macro-average of the ROC curve is 0.83.

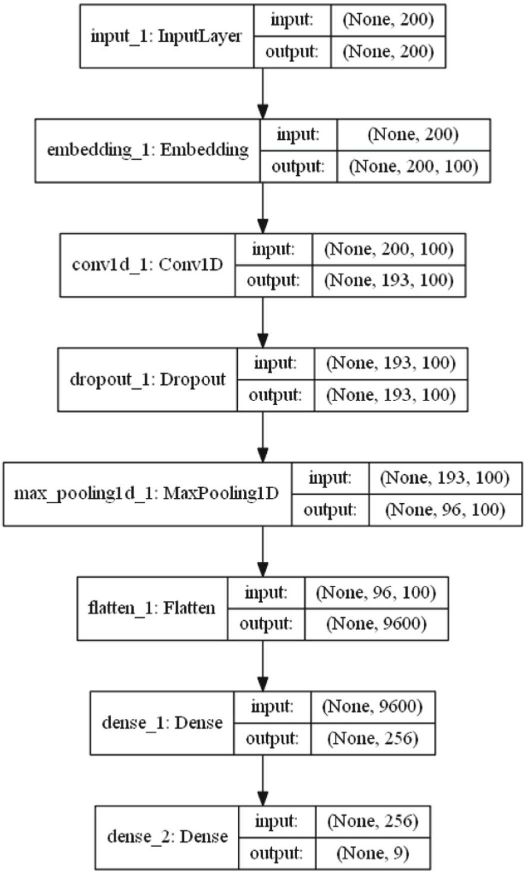

Fig. 3. A model with 100 filters and eight region size

Effect of Multiple Region Size: Here we considered the effect of 2, 3, and 4 regions.

1. **Effect of 2 regions with different size**
   Here we used 50 filters for each region, and experimentation is done for two regions of different sizes. We experimented with two regions of different sizes, as shown in Table 4. Accuracy on validation data is 0.67 for (3,5) region size. Two channels are built one for each region size, same as that of 1 region model. The output from two channels is concatenated before it is given to a dense layer.
   Table 5 shows classwise precision, recall, f1-measure for (3,5) region model. It shows that a maximum f1-score of 0.79 is obtained for 'Bhakti' type of classes. The average precision score, micro-averaged over all classes for 2 regions (3,5): 0.70.

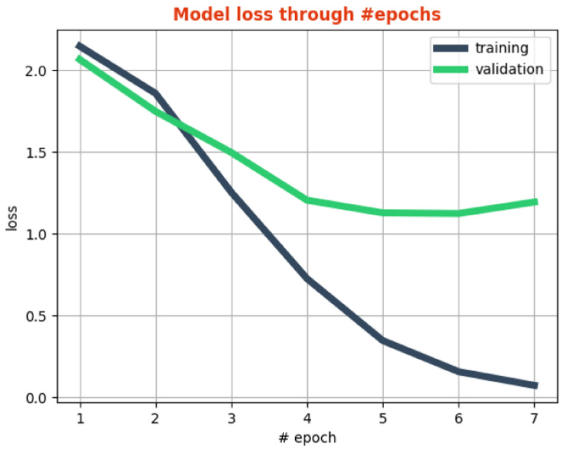

Fig. 4. Model loss through epochs for the model shown in Fig. 3

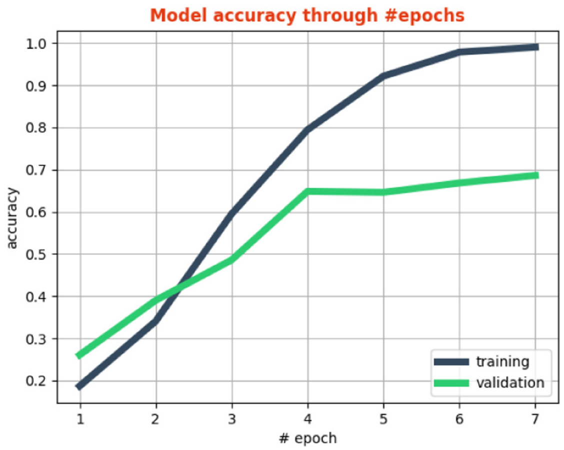

Fig. 5. Model accuracy through epochs for the model shown in Fig. 3

Figure 8 shows the confusion matrix for (3,5) region. It shows that maximum poems of 'vidamban' type of category get misclassified into 'gambhir' and 'vinodi' type of category. Also, maximum of 'Shrungar' types of poems gets misclassified into 'prem' category. Micro-average ROC curve is 0.83, and the macro-average ROC curve is 0.82 for the (3,5) model.

2. **Effect of 3 regions with different size**
Here we experimented with 3 regions of different sizes with 32 filters for each region. For three regions, three channels are created, which are then concatenated. The model for each channel is the same as the model for one region, except 32 filters are used for each region. Convolutions are performed using 32 filters for three words for the region of size 3. Similarly, for four

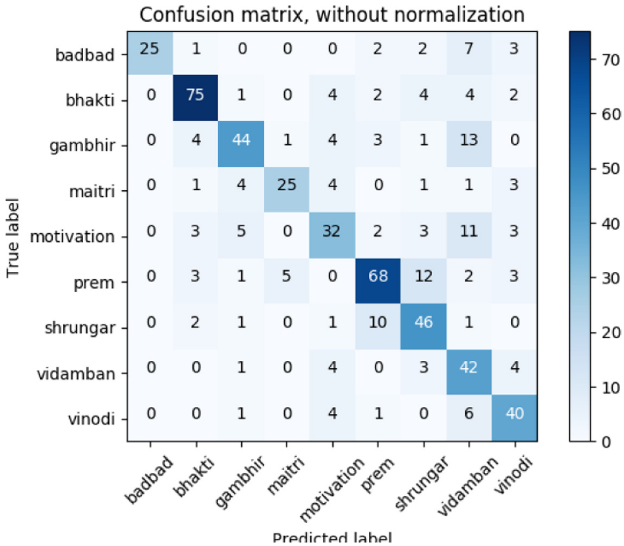

Fig. 6. Confusion matrix for the model shown in Fig. 3

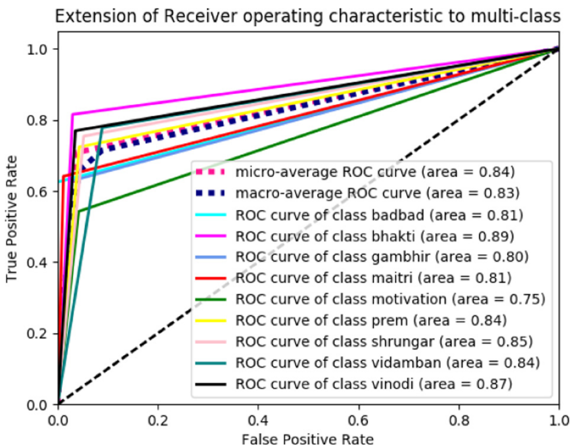

Fig. 7. ROC curve for the model shown in Fig. 3

regions, four words and for five regions, five words filtering is done. The highest accuracy of 69% is achieved for the three regions as shown in Table 6.

3. **Effect of 4 regions with different size**
   We experimented with four regions of different sizes with 32 filters for each region. Table 7 shows that maximum accuracy of 0.71 on validation data and 0.73 on test data is obtained for four regions (4, 5, 6, 7). Table 8 shows

Table 3. Classwise precision, recall, f1-measure for 8 region model.

Class type	Precision	Recall	f1-score	support
badbad	1.00	0.62	0.77	40
bhakti	**0.84**	**0.82**	**0.83**	92
gambhir	0.76	0.63	0.69	70
maitri	0.81	0.64	0.71	39
motivation	0.60	0.54	0.57	59
prem	0.77	0.72	0.75	94
shrungar	0.64	0.75	0.69	61
vidamban	0.48	0.78	0.60	54
vinodi	0.69	0.77	0.73	52
avg/total	**0.73**	**0.71**	**0.71**	561

Table 4. Effect of 2 regions.

Region sizes	Accuracy on validation data	Accuracy on test data
(2,4)	0.66	**0.70**
(3,5)	**0.67**	0.69
(4,6)	0.66	0.69
(5,7)	0.66	0.68
(6,8)	0.66	**0.70**

Table 5. Classwise precision, recall, f1-measure for (3,5) region model.

Class type	Precision	Recall	f1-score	support
badbad	**0.89**	0.69	0.78	45
bhakti	0.76	**0.82**	**0.79**	97
gambhir	0.74	0.55	0.63	64
maitri	0.71	0.77	0.74	39
motivation	0.38	0.57	0.46	49
prem	0.60	0.72	0.66	80
shrungar	0.85	0.59	0.69	56
vidamban	0.73	0.62	0.67	77
vinodi	0.72	0.76	0.74	54
avg/total	**0.71**	**0.68**	**0.69**	561

classwise precision, recall, f1-measure for four regions model of size (4, 5, 6, 7). Figure 9 shows the confusion matrix for the model with four regions of size (4,5,6,7).

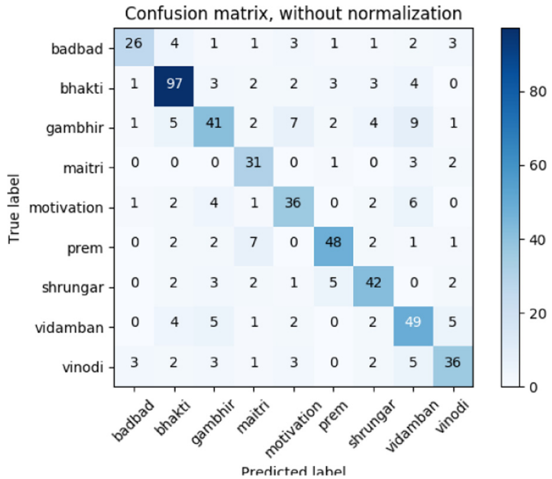

Fig. 8. Confusion matrix for the (3,5) region model

Table 6. Effect of 3 regions.

Region sizes	Accuracy on validation data	Accuracy on test data
(2,3,4)	0.66	0.68
(2,4,6)	0.69	**0.69**
(3,4,5)	0.68	0.68
(4,5,6)	0.67	0.68
(4,6,8)	0.67	**0.69**
(5,6,7)	0.68	**0.69**
(6,7,8)	**0.71**	0.68

Table 7. Effect of 4 regions.

Region sizes	Accuracy on validation data	Accuracy on test data
(2,3,4,5)	0.69	0.65
(3,4,5,6)	0.69	0.71
(4,5,6,7)	**0.71**	**0.72**
(5,6,7,8)	0.68	0.71

Effect of Different Batch Sizes: Here we experimented with different batch sizes of 16, 32, 64 and 128, considering three regions 4, 6 and 8 with 32 filters for each region. Table 9 shows the results for different batch sizes.

Table 8. Classwise precision, recall, f1-measure for (4, 5, 6, 7) regions model.

Class type	Precision	Recall	f1-score	support
badbad	0.81	0.62	0.70	42
bhakti	**0.82**	**0.84**	**0.83**	**115**
gambhir	0.66	0.57	0.61	72
maitri	0.65	0.84	0.73	37
motivation	0.67	0.69	0.68	52
prem	0.80	0.76	0.78	63
shrungar	0.72	0.74	0.73	57
vidamban	0.62	0.72	0.67	68
vinodi	0.72	0.65	0.69	55
avg/total	**0.73**	**0.72**	**0.72**	**561**

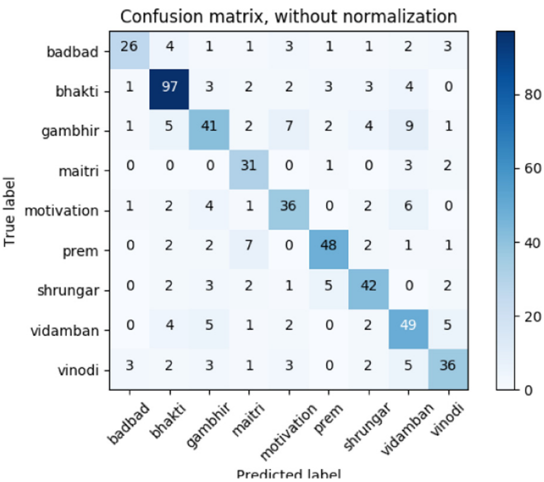

Fig. 9. Confusion matrix for the model shown in model of 4, 5, 6, 7 regions

Table 9. Effect of different batch size.

Batch size	Accuracy on validation data	Accuracy on test data
16	0.67	0.70
32	0.69	0.71
64	**0.66**	**0.73**
128	0.70	0.69

4 Conclusion

Here we used CNN based deep learning models for the classification of Marathi poems. In this approach, we experimented with one region, 2, 3, 4 regions for filtering. For 1-region size filter, maximum accuracy 0.69 on validation, and 0.71 for test data is obtained for 8-region size. For 2-regions, filter maximum accuracy 0.66 on validation and 0.70 for test data is obtained for (2,4) and (6,8) regions. For 3-regions, filter maximum accuracy 0.69 on validation and 0.69 for test data is obtained for (2, 4, 6) regions. For 4-regions filter maximum accuracy 0.71 on validation and 0.72 for test data is obtained for (4, 5, 6, 7) regions. For different regions, results show that maximum accuracy achieved is 0.72. In CNN, we also experimented for different batch sizes for three regions (4, 6, 8), each with 32 filters. Results show that for a batch of 64, the maximum accuracy on test data achieved is 0.73. All the above models show that a maximum accuracy of 0.73 is achieved for the classification of 2803 poems into nine categories. Our models can be used for essay classification, news classification, story classification, etc. In the future, we can increase the dataset, tune different parameters like batch size, activation function, optimizers, etc. Future work can be focused on the use of Recurrent neural network, Long short-term memory, auto encoder-decoder models, pretrained models, attention based models for Marathi text classification.

References

1. Gu, J., et al.: Recent advances in convolutional neural networks. Pattern Recogn. **77**, 354–377 (2018)
2. Kowsari, K., Jafari Meimandi, K., Heidarysafa, M., Mendu, S., Barnes, L., Brown, D.: Text classification algorithms: a survey. Information **10**(4), 150 (2019)
3. Minaee, S., Kalchbrenner, N., Cambria, E., Nikzad, N., Chenaghlu, M., Gao, J.: Deep learning-based text classification: a comprehensive review. ACM Comput. Surv. (CSUR) **54**(3), 1–40 (2021)
4. Kamath, C.N., Bukhari, S.S., Dengel, A.: Comparative study between traditional machine learning and deep learning approaches for text classification. In: Proceedings of the ACM Symposium on Document Engineering, p. 14 (2018)
5. Georgakopoulos, S.V., Tasoulis, S.K., Vrahatis, A.G.: Convolutional neural networks for toxic comment classification. In: Proceedings of the 10th Hellenic Conference on Artificial Intelligence, p. 35. ACM (2018)
6. Cano, E., Morisio, M.: A deep learning architecture for sentiment analysis. In: Proceedings of the International Conference on Geoinformatics and Data Analysis, pp. 122–126. ACM (2018)
7. Hughes, M., Li, I., Kotoulas, S., Suzumura, T.: Medical text classification using convolutional neural networks. Stud. Health Technol. Inform. **235**, 246–250 (2017)
8. Hsu, T., Tzuhan, Y.: Petroleum engineering data text classification using convolutional neural network based classifier. In: Proceedings of the 2018 International Conference on Machine Learning Technologies, pp. 63–68. ACM (2018)
9. Kalchbrenner, N., Grefenstette, E., Blunsom, P.: A convolutional neural network for modelling sentences. arXiv preprint arXiv:1404.2188 (2014)

10. Kim, Y.: Convolutional neural networks for sentence classification. arXiv preprint arXiv:1408.5882 (2014)
11. Zhang, Y., Wallace, B.: A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification. arXiv preprint arXiv:1510.03820 (2015)
12. de Sousa Pereira Amorim, B., Alves, A.L.F., de Oliveira, M.G., de Souza Baptista, C.: Using supervised classification to detect political tweets with political content. In: Proceedings of the 24th Brazilian Symposium on Multimedia and the Web, pp. 245–252. ACM (2018)
13. Severyn, A., Moschitti, A.: UNITN: training deep convolutional neural network for twitter sentiment classification. In: Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015), pp. 464–469 (2015)
14. Baker, S., Korhonen, A., Pyysalo, S.: Cancer hallmark text classification using convolutional neural networks. In: Proceedings of the Fifth Workshop on Building and Evaluating Resources for Biomedical Text Mining (BioTxtM 2016), pp. 1–9 (2016)
15. Ali, A.R., Ijaz, M.: Urdu text classification. In: Proceedings of the 7th International Conference on Frontiers of Information Technology, p. 21. ACM, December 2009
16. Krail, N., Gupta, V.: Domain based classification of Punjabi text documents using ontology and hybrid based approach. In: Proceedings of the 3rd Workshop on South and Southeast Asian Natural Language Processing, pp. 109–122 (2012)
17. Rajan, K., Ramalingam, V., Ganesan, M., Palanivel, S., Palaniappan, B.: Automatic classification of Tamil documents using vector space model and artificial neural network. Expert Syst. Appl. **36**(8), 10914–10918 (2009)
18. Patil, J.J., Bogiri, N.: Automatic text categorization: Marathi documents. In: 2015 International Conference on Energy Systems and Applications, pp. 689–694. IEEE (2015)
19. Deshmukh, R.A., Kore, S., Chavan, N., Gole, S., Kumar, A.: Marathi poem classification using machine learning. Int. J. Recent Technol. Eng. (IJRTE) 2723–2727 (2019). ISSN 2277–3878
20. Ahmad, S., Asghar, M.Z., Alotaibi, F.M., Khan, S.: Classification of poetry text into the emotional states using deep learning technique. IEEE Access **8**, 73865–73878 (2020)
21. O'Shea, K., Nash, R.: An introduction to convolutional neural networks. arXiv preprint arXiv:1511.08458 (2015)
22. Wu, J.: Introduction to convolutional neural networks, vol. 5, no. 23, p. 495. National Key Lab for Novel Software Technology. Nanjing University, China (2017)
23. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
24. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from overfitting. J. Mach. Learn. Res. **15**(1), 1929–1958 (2014)