CNN Classification Approach For Analysis And Recognition Of Marathi Manuscript

Sarika T. Deokate, Nilesh J. Uke

Abstract: Day by day, the printed and handwritten documents are processed on high scale to perform different operations digitally. In this process, Classification of the contents is one of the crucial tasks for any Optical Character Recognition. Diverse techniques like SVM,KNN, ANN, Deep Learning and many more has been utilized for this. We have used KNN, CNN approach for this and illustrated the working of the Convolutional neural network for the Marathi Script with discussion of its working environment. To achieve the superiority we used the max pooling. We tried our system on the diverse size dataset to check the fitness of the proposed model. We evaluated our system on the varying size trained and test data set

Index Terms: Deep Learning, Convolution Neural Network, Document Classification, OCR, Pooling.

1. INTRODUCTION

Classification is used for identification of Text letters, numerals and other symbols if any. In this phase, the different parameter also gets checked like PSNR, false rate, precision. To provide the data to the post-processing phase, precision of the classification must be superior. If the characters identified are erroneous, then it will be interpreted wrongly and there will be mismatch of the letters in particular word. Therefore success of any OCR is dependent on the précised classification[19],[4]. To perform the précised categorization, the proficient feature excerption approaches need to be utilized. These features will be utilized to categorize the letter accurately[6],[12]. To perform the accurate identification and categorization, the huge dataset is required in any image, video or any categorization. Without proper dataset, it is not possible to outperform the categorization for any OCR. Many researchers designed the dataset for the printed, handwritten manuscripts previously [6], [10]. SVM, ANN, KNN, NN, Deep learning are some of the approaches broadly used for the categorization. ANN and deep learning provided the superior result for the distinct script [1],[16],[17] . Author evaluated the performance of the SVM and ANN and ANN outperformed well as compared to SVM [11].

2 RELATED WORK

As India is a multiple language country, many researchers concentrated on the distinct languages of India for handwritten character identification. Due to variance of the number of the aspects of handwritten content, it is bit difficult to perform the character level identification truthfully. [14] Pal et. al. has designed a model, which is utilized to identify the six distinct handwritten numerals of Indian scripts. Here the amended quadratic categorization is utilized to perform the identification of these distinct scripts i.e. Oriya, Telugu, Bangla, Devnagari and Tamil. Directional knowledge is utilized for the aspect identification in categorization. Bounding box scheme is utilized to perform the fragmentation and directional aspects are estimated. Miciak offered the technique for character detection. Here author utilized the numeral, various characters utilized on postal transcripts e.g. postal codes, addresses of mail section [13]. Arora et.al. has used four feature excerption techniques for hand written Devnagari character identification i.e. shadow feature, intersection, straight line fitting technique and chain code histogram. The multiple classifier grouping has been utilized for handwritten offline character identification [2]. [7] designed an approach for identification of Devnagari script. Author has prepared the dataset of handwritten and utilized the ISM standard font of printed characters. 3000 trials of numerals and 4500 character samples are utilized in the form of grey levels. To test the performance, NN, reduced NN, KNN Euclidian distance based KNN and other similar variant classifiers are utilized. Unconstrained manuscript handwritten Marathi characters was utilized by Shelke & Apte [18]. [15] presented technique for Text-line excerption for handwritten manuscripts. However, most of researchers exploited the aspects of certain languages and worked only for a specified language. Wang et.al. intended a topic language model amendment mode to obtain the superior discovery performance for the Chinese transcript which is homologous and offline handwritten images. The method is demonstrated on homologous Chinese offline handwritten transcript image identification [21]. To solve the composite problems, the machine learning conceptions are adapted rapidly [8].

Deep learning is the blazing idea utilized in the computer vision. Deep learning does not need the individual extraction of feature as normal machine learning systems do. It performs the extraction of the image features as the one of the built-in functionality of the categorization. CNN model is intended in such a ways that it acclimatizes the multilayer perception and necessitated very limited preprocessing. It works like a biological process of animals, which share the interconnected pattern of the neuron. Utilizing the convolution, the higher-level features of the data is aimed in CNN. It is extensively used in the computer vision, graphics, medical, space, and lot of applications[17]. Case study has been illustrated to check the working capacity of the Deep learning conception. This work has been performed on the series of handwritten letters. To examine the performance the five diverse depths has been checked for the CNN convolutional neural network. Two diverse dataset have been utilized in this study. 49 set of character classes in one dataset and 47 set of character classes have been established. Trained and set of test dataset has been prepared for the categorization purpose. Extraction of the features is performed utilizing the HOG[13]. In [20], Tang

- *Dr. Sarika T. Deokate, Associate Professor, Department of Computer Engineering, SBPCOE Indapur Pune India.*
- *Dr. Nilesh J. Uke, Principal, Department of Computer Engineering, TAE, Pune, India Email: nileshjuke@gmail.com*

worked on both softmax function and designed the DLSVM i.e. deep learning SVM. In this study, the softmax function is substituted by the designed DLSVM. Researcher has proved that the DLSVM is better option than the softmax function for the categorization for their utilized datasets. They utilized the CIFAR-10, MNIST dataset for this work for face expression identification. This work was offered in the ICML 2013 workshop. For this, 28709 images with 48 by 48 size has been utilized underneath 7 diverse expression style.

3 PROPOSED CONVOLUTIONAL NEURAL NETWORK

We utilized the sequential model of the Keras to execute the CNN. Very essential part of this system is to choose the model. It is linear channel or stack of available neural network levels. In this work, initially we imported the sequential model of the Keras. Every node of this model is instantiated with the certain weights. Weights are instantiated with the diverse techniques; one is random uniform in which arbitrary tiny values are initiated. Second is random_normal; where weights are instantiated as per the Gaussian including the tiny standard variance and zero mean.
At the time of actual implementation, this model includes the following stages:

- Identify the input and output.
- Designing of the model with dimensions.
- Define the architecture and construct the computational graph.
- Indicate the configuration of the learnable procedure and optimizer.
- Need to train and then perform the testing on the given specified dataset.

The standardized procedure to perform the categorization using CNN model is as beneath:

- Performing the pipelining or stacking by utilizing the add() mode.
- Configuration of the learnable procedure utilizing the compile mode with particular parameters.
- Fit mode is utilized for grounding the training procedure

3.1 Grounding of the Model

The source input is provided to first level in the provided network. Every layer/level acquires the input, which accomplish an activation function and obtains the activation map. It is provided to subsequent levels until we get outcome from the very last level/layer. As this is a straight forward conception, it is termed as Forwarding-Pass approach[9]. Second approach utilized in this structure is back propagation-pass. Here the training of the provided data is performed in frequentative. We can adapt the distinct network parameters at every other iteration to boost the superiority of the outcome. At preliminary state of the process, the network is unaware about current input data to be treated as parameters. In forwarding-pass, there may be chances of getting false outcome. So it is essential to present the error metric i.e. cost/loss. The loss can be estimated by checking out the loss function on expected outcome and then forecasted the trained outcome. So now, at learnable states, model can settle on the alteration of the utilized parameters to lessen the loss. This can be performed in frequentative manner to decide the final parameter of the model. This can be further tried for the subsequent exemplars and then we can choose the learning score based on acquired negative gradient. This fraction is included to renew the weights. Utilizing this whole process, it tends to provide the optimal set of learnable network parameters and confluence to its local minima. And this is termed as SGD i.e. stochastic gradient n/w parameters. The SGD function is depicted as

$$Par_{k+1} = Par_k - \alpha \nabla Par_k \, Ls(Par_k)$$

Where, Par_{k+1} are fresh learnable parameters.

Par_k are present parameters, α is the proportion of learning, ∇Par_k is the gradient w.r.t. Par_k and loss function $Ls(Par_k)$.

As we have utilized the Keras model, the process of Keras illustrated in short here. To utilize the model, it is necessary to specify the input profile. So it is essential to supply the shape information at the commencing stage only. In subsequent phases, it is not necessary to supply the shape data, it may take it automatically. Certain 2D level specifies the input profile information via dense level as parameters like input dim. Even a constant batch dimensions can utilized in several situations. Prior to commencing the model training, it is necessary to build up the learnable procedure. This can be done through, making the use of compile procedure. It takes the set of metrics for the categorization purpose. Its value can be “accuracy”. Next parameter utilized here is optimizer which specifies the kind of optimizer utilized e.g. Adagrad, rmsprop, Adam, SGD (stochastic gradient optimizer), Adamax. Many more optimizer are available which is utilized to work out with optimization problems. There can be a problem to discover the optimal solution or poor learning proportion/rate or improper scaling of the dataset. These or these types of other problem may decrease the overall working environment of the model. It is required, to provide the optimizer to have the solution to these problems and to boost the performance. Models adapted the Kera are normally trained with numpy arrays input source data and provided labels. Then fit function will be utilized. Generally, variety of procedures with its variety of attributes is utilized in these models. To train a model, it needs to be set with the proper dataset. Training the small size images can be efficiently handled by the many conventional categorizers. But to handle the large set of images, deep learning model can be utilized significantly. Deep learning conception works well with the feature extraction approach. Many times, it discovers and chooses the fascinating features which are not specified manually or which were not available before. To achieve this, we need to provide the large enough samples in the training dataset. This needs the relevant size, dimensions and deepness of the network. CNN adapts translation-invariance, local features.

3.2 Preparation and Configuration of Dataset for CNN

For any model, it is essential to prepare the source data into the relevant form for the further processing. In this model, data is configured to tensors of floating-point prior to providing it to the modeled network. To perform this, images are retrieved one by one. Then transformed it, to the pixel grid of tensors, which are in floating-point from. Further these images are remapped to the range of [0,1]. Then this will be reformed to

an array of 28 by 28 in linear way. Instead of utilizing the images directly from the dataset, the CNN perform the task of rearranging these dataset which will fit with the designed model. This is performed, by doing the operation on the dataset images repetitively to create enhanced image information required for the model. It may increase the processing rate, but diminishes the memory usage. It performs operations like revolve, resize, flips, shrivel, zooming etc. to fit in the model. Statistical estimation will be performed. Over-fitting may occur due to small number of data samples available in the dataset and is not able to perform the training properly to simplify the new data. With the extent of the huge dataset, it may include all the possible aspects available, to avoid the over-fitting. Data augmentation is one of the approaches utilized in the deep learning to enhance the number of samples in the training from the previously available data samples. This system builds the new data set by making the arbitrary alterations in the previous available data samples. This is carried out to ensure and access all the possible aspects and to oversimplify the process of categorization. This fitting will be performed in epoch process. One epoch performs the fetching of dataset frontward and rearward just the once through provided network. But it's a tedious task to perform the one epoch, so it is segregated in tiny size batches. We are fetching the intact dataset but in batches. The weights are altered in the provided network for every augmenting epoch. The curve may transform from under-fitting to optimal one and then may be over-fitting one. Moreover, that is why we need to finalize the number of performed epoch in the network. Iteration is nothing but the amount of batches necessitated to finish one epoch. E.g. if we have 5000 of dataset samples with batches of 1000, then it requires 5 iteration to finish the 1 epoch. Fitting generator is utilized and validation data will be passed to take out for the assessment. The dataset for the validation is provided to carry out the nonpartisan assessment of the model to fit on the trained dataset. These samples are utilized to fine tune the hyper-parameters. The test dataset and set of validation is distinct. Testing dataset is utilized to check the assessment of the model. Once the model is properly trained, testing will be evaluated. Several times, set of validation can be utilized as a testing one. To perform this task, the dataset can be segregated into testing and training dataset and some data can be utilized for the validation purpose. Here the image is depicted as the 1D array of 28 by 28 dimension (784) of float value amid 0 to 1.

4 PROJECTED MODEL DISCUSSIONS

In our system, we utilized the three conv2D with two maxpooling2D phases. This performs better to enhance the competence of the network and also tries to diminish the amount of the feature map to avoid the overlying at the phase of Flatten layer. The deepness of the feature map gradually enlarges the network commencing 32 towards the 128 and the extent of the feature maps diminishes from 28 by 28 towards 3 by 3. The whole procedure is illustrated beneath

4.1 Working of Convolution Procedure

Convolution layer maps the local aspects of the images. These images can be split into number of local aspect such as color form, edges, channels etc. The learnable aspects are not adaptable to transformation invariance. These aspects can be recognizable at any time in CNN convolution. But in densely associated network, it learns the aspects again every time at some new position. That is why; CNN is proficient for the images. The CNN adapts the aspects of the images in hierarchical manner. Means the very first layer of convolution level adapts the tiny aspects in the beginning and at next level, it learns the subsequently larger aspects from the previous layer and performs this so on. Thus, it tries to learn the more complex aspects gradually. The layers performed in this work are illustrate below

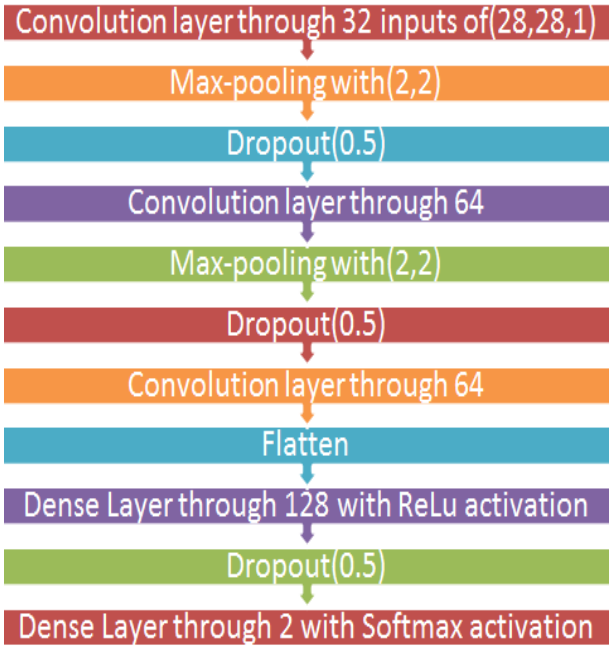

Figure 1. *Proposed System*

It works above the 3 aspects termed as, feature map including height, breadth and also the number of channels i.e. depth. In colored, i.e. RGB form picture it have three palettes in it so depth will be fixed to 3. For the Bi-level pictures, the depth will be set to 1. This depth can be set to be subjective, as this is the parameter of the particular layer and at individual layers, they take charge as filters. A filter performs the encoding of the certain aspects of the source input. In our system the foremost convolution level captures a feature map to the extent of 28 by 28 i.e. (28, 28, 1) and extends feature map of dimensions of (26 ,26 ,32) for the output. At the foremost level, it is also necessary to provide the size of the input. It estimates total 32 filters above source input. Those 32 resultant output channels contain 26 by 26 of reticulation of values. This is nothing but the outcome of particular filter aspects at distinct positions in the source input and this is termed as the response map of particular filter mapped above the source input. In estimated feature map, the size in the axis of depth is nothing but a filter (feature) and other two entities are the spatial map of the occurred response by this filter above the source input. The main parameters utilized by the convolutions are depth, provided for the feature map of the output and second, the quantity of the patches taken out from the source input. The volume of the filters is estimated by the utilized convolution. E.g. in our system, it is commenced at the deepness of 32 and then concluded with 64 depth. Patches of the features are generally 5 by 5 or 3 by 3. Mostly 3 by 3

patches are utilized. We also utilized the 3 * 3 patch. These windows of 3 by 3 or 5 by 5 sizes will be slithered in the convolution state above the 3 dimensional inputted feature maps. It evaluates each potential position and extricated the 3D piece of adjoining features i.e. outline with height and width of the window with its deepness. Then every patch/piece in 3D form is transmuted to a vector form of 1D and it is also termed as convolution kernel. Moreover, every one of those vectors is spatially correspondent into the output map of 3D form with the outline of height, width and output deepness. The size of the output may be differed from the input form, due to the utilization of the strides and due to the consequences of the border, which gets occurred due to the input feature map padding. These effects are illustrated here. Let us consider a 5 by 5-feature map and so there will be merely total 9 surface in which we can concentrate for a centroid of 3 by 3 window by taking the grid of 3 by 3. It is necessary to validate this piece or grid in the 5 by 5-source feature map. The outcome feature map will be of size 3 by 3. It condenses by means of 2 tile sizes beside all the dimensions available. This is what the effect gets transformed when we shift the convolution from the 28 by 28 size input towards 26 by 26 size in the subsequent phase or layers of convolution. Therefore, if we require the same size of spatial dimension for the outcome feature map, then we have to perform the padding conception. Padding comprises a procedure, which adjoins the suitable quantity of rows with required columns beside every plane of the source feature map. The strides are also taken into consideration at the convolution level and it is one of customary factor need to be specified which affects the area of the outcome feature map. Normally, the central patches/pieces of the convolutional window are mostly adjoining. The space amid the two consecutive windows is a one of the parameter in the convolution, which is nothing but a stride and set to 1 by defaulting. More than 1 stride can be utilized, if required.

Here the exemplar is illustrated for the 3 by 3-convolution tile utilizing 2 by 2 strides.

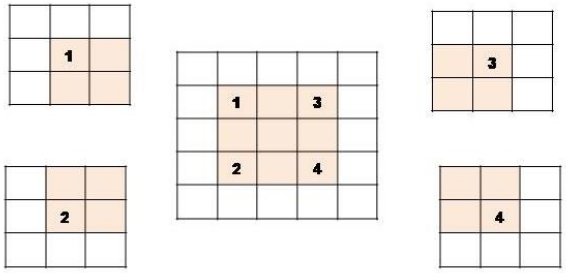

Figure 2. *Stride Representation*

In this Figure 2, the tiles are taken out without padding with a 3 by 3 convolution utilizing the stride of 2 above the 5 by 5 source.

ReLu- Activation in Convolution Layer

Rectified Linear unit outperforms in terms of 0, if the provided input lessen than 0 or it will provide the raw output. If we provide the input value more than 1, then produced output is identical with the provided input.

$$fn(x_i) = \max(0, x_i)$$

This termed as, if the input xi > 0 then the outcome will be xi and if xi < 0 then outcome will be 0.

These activations are non-linear function. Means when we obtain the positive input, the obtained derivative is also 1. It doesn't perform any condense effect. It also trains the model in very rapid manner(Satya Mallick, n.d.). The behavior of ReLu is illustrated in figure 3 and 4.

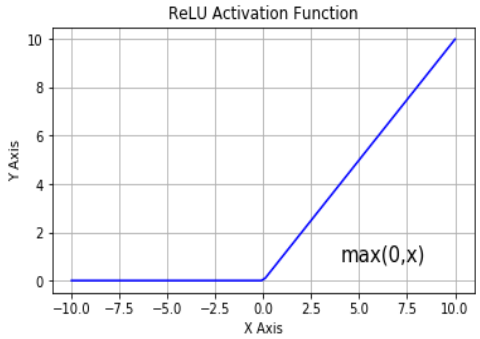

Figure 3*. ReLU Activation Function (Satya Mallick)*

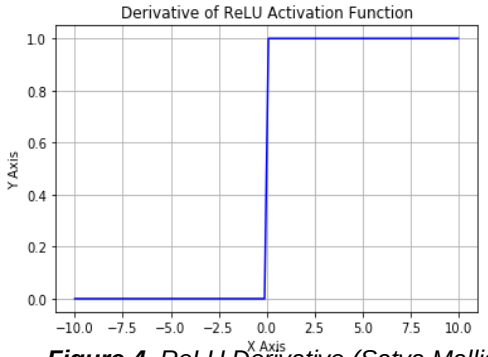

Figure 4*. ReLU Derivative (Satya Mallick)*

4.2 Working of Max Pooling Layer

In the CNN models, the area of the feature map is bisected after performing the MaxPooling2D level. It is utilized instantly, following the convolution level to diminish the spatial dimensions of the map. Means as in this system, if the first feature map was of 26 by 26 sizes, then it gets bisected to 13 by 13 size. It performs down sampling of the feature map. The max-pooling is performed with 2 by 2 window size with stride of 2. Max-pooling is performed to accomplish the down-sampling, which is necessary to discover and shun the over-fitting at feature mapping. Instead of max pooling, we can also utilize strides as we illustrated previously or even average pooling approach. However, max pooling is fitted to be good as compared to other options. Instead of checking the average mapping, it is better to check max behavior of the points in the tile. Hence, in this approach, dense map for the features is first extracted and then activation map has been checked at tiny patches. Dropout has been performed after execution of every max pooling. In this system, provided the rate of drop to execute the dropout function with value of .5. It is utilized to shun the over-fitting. The rate value can be set amid 0 and 1. In the experimentation, we tested the system with the distinct values of the dropout and then utilized the optimal value for the current work. Flatten is done after three layer of convolutions. Flatten performs the transformation 3D or 2D

form to 1D form with conserving the ordering of the weights. Then we adjoined the dense layer with 128 values. Dense layer is nothing but the fully associated level of the CNN. Two dense level are provided with the output size specification. Once the model is prepared it is compiled and trained utilizing the fit option. Thus, we studied and designed the CNN in work and tested rigorously on the dataset, which we prepared. We tried procedure with the softmax activation fully associated levels and utilized the RmsProp optimizer with the 128 batch size. Then, we got the little over-fitting but with the superior accuracy. Here we utilized the 3600 dataset. The loss occurred due to the training contents are diminishing with every epoch. However, accuracy precision of training contents is boosting with every epoch.

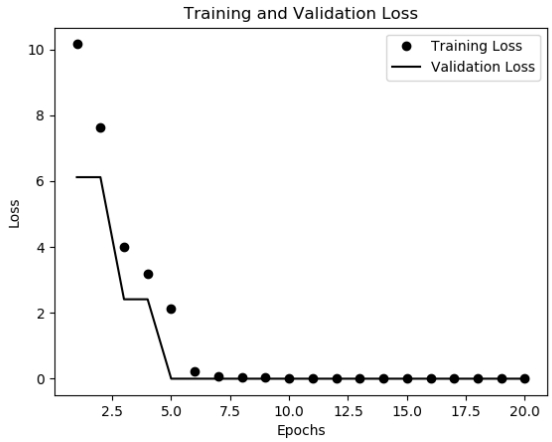

***Figure 5**. Loss Outcome*

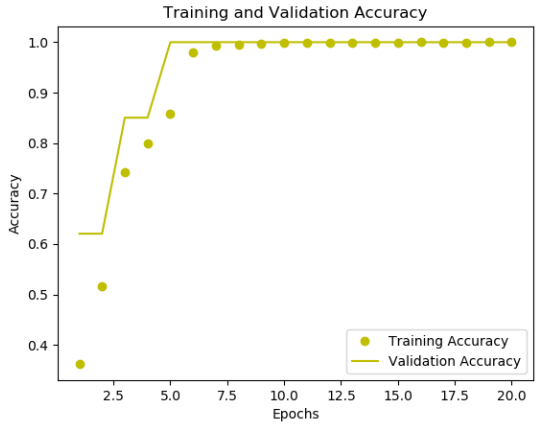

***Figure 6.** Accuracy Outcome*

We tried our system for the varying size of dataset and got the good outcome.

5 CONCLUSION

We demonstrated the deep learning CNN model. Again, here before designing and performing the actual model, we first grounded our dataset to absolutely fit for our system. The augmentation approach has been utilized to boost the dataset. We prepared our own dataset in this work. In this part, we illustrated the working of the convolutional network with its distinct levels. We utilized the Convolutional level, max pooling and Dense i.e. fully associated level to put this model into practice. Here we discussed about the parameters utilized and the outcome of our system.

REFERENCES

[1] Acharya, S., Pant, A. K., & Gyawali, P. K. (2015). Deep Learning Based Large Scale Handwritten Devanagari Character Recognition. 2015 9th International Conference on Software, Knowledge, Information Management and Applications (SKIMA) Deep

[2] Arora, S., Bhattacharjee, D., Nasipuri, M., Basu, D. K., & Kundu, M. (2008). Combining Multiple Feature Extraction Techniques for Handwritten Devnagari Character Recognition. Industrial and Information Systems, 2008. ICIIS 2008. IEEE Region 10 and the Third International Conference On, 1–6. https://doi.org/10.1109/ICIINFS.2008.4798415

[3] Chakraborty, B., & Shaw, B. (2018). Does Deeper Network Lead to Better Accuracy : A Case Study on Handwritten Devanagari Characters. 2018 13th IAPR International Workshop on Document Analysis Systems (DAS), 411–416.

[4] Deokate S., Uke N. (2018) Various Traditional and Nature Inspired Approaches Used in Image Preprocessing. In: Pawar P., Ronge B., Balasubramaniam R., Seshabhattar S. (eds) Techno-Societal 2016. ICATSA 2016. Springer, Cham

[5] Dongre, V. J., & Mankar, V. H. (2012). Development of Comprehensive Devnagari Numeral and Character Database for Offline Handwritten Character Recognition. Applied Computational Intelligence and Soft Computing, 2012, 1–5.

[6] Dongre, V. J., & Mankar, V. H. (2013). Devnagari Handwritten Numeral Recognition using Geometric Features and Statistical Combination Classifier. International Journal on Computer Science and Engineering (IJCSE), 5(10), 856–863.

[7] Holambe, A. N., Holambe, S. N., & Thool, R. C. (2010). Comparative study of devanagari handwritten and printed character & numerals recognition using Nearest-Neighbor classifiers. Proceedings - 2010 3rd IEEE International Conference on Computer Science and Information Technology, ICCSIT 2010, 1, 426–430.

[8] Indolia, S., Goswami, A. K., Mishra, S. P., & Asopa, P. (2018). Conceptual Understanding of Convolutional Neural Network- A Deep Learning Approach. Procedia Computer Science, 132, 679–688.

[9] Ian Goodfellow Yoshua Bengio, & Courville, A. (2016). Deep Learning. Retrieved from http://www.deeplearningbook.org

[10] Jayadevan, R., Kolhe, S. R., Patil, P. M., & Pal, U. (2011). Database development and recognition of handwritten Devanagari legal amount words. Proceedings of the International Conference on Document Analysis and Recognition, ICDAR, 304–308.

[11] Kamble, P. M., & Hegadi, R. S. (2015). Handwritten Marathi character recognition using R-HOG feature. Procedia Computer Science, 45(C), 266–274.

[12] Kavallieratou, E., Fakotakis, N., & Kokkinakis, G. (2002). Handwritten character recognition based on structural characteristics. Object Recognition Supported by User Interaction for Service Robots, 3, 139–142.

[13] Miciak, M. (2008). Character recognition using radon transformation and principal Component analysis in postal applications. Proceedings of the International Multiconference on Computer Science and Information Technology, IMCSIT 2008, 3, 495–500. https://doi.org/10.1109/IMCSIT.2008.4747289

[14] Pal, U., Wakabayashi, T., Sharma, N., & Kimura, F. (2007). Handwritten numeral recognition of six popular Indian scripts. Proceedings of the International Conference on Document Analysis and Recognition, ICDAR, 2, 749–753.
[15] Ryu, J., Koo, H. Il, & Cho, N. I. (2014). Language-independent text-line extraction algorithm for handwritten documents. IEEE Signal Processing Letters, 21(9), 1115–1119.
[16] Sarika T. Deokate, Nilesh J.(2019a) Uke Devnagari Script Categorization by Utilizing CNN and KNN International Journal of Innovative Technology and Exploring Engineering (IJITEE)ISSN: 2278-3075, Volume-8 Issue-5 March, 2019a
[17] Sarika T Deokate, Nilesh J Uke, Deepak S Dharrao Review on Deep Learnable Approach for Categorization JASC: Journal of Applied Science and Computations 6 (2) pp. 2530-2533 2019b
[18] Shelke, S. S., & Apte, S. S. (2011). A Multistage Handwritten Marathi Compound Character Recognition Scheme using Neural Networks and Wavelet Features. … and Pattern Recognition, 4(1), 81–94.
[19] Shih, F. Y. (2010). Image Processing and Pattern Recognition. John Wiley & Sons, Inc., Hoboken, New Jersey. https://doi.org/10.1002/9780470590416
[20] Tang, Y. (2013). Deep Learning using Linear Support Vector Machines. https://doi.org//S0102-311X2006000600002
[21] Wang, Y., Ding, X., & Liu, C. (2014). Topic Language Model Adaption for Recognition of Homologous Of fl ine Handwritten Chinese Text Image, 21(5), 550–553.
[22] Satya Mallick. “https://www.learnopencv.com/understanding-activation-functions-in-deep-learning/”