Journal of Computing & Biomedical Informatics
ISSN: 2710 - 1606

Volume 02 Issue 01
2021

Research Article

Image Classification and Text Extraction using Convolutional Neural Network

Rabia Zaman[1], Rafia Bashir[1] and Atif Raza Zaidi[2]

[1]Department of Computer Science, Lahore Garrison University, Lahore, 67000, Pakistan
[2]The Faculty of Information Technology (FOIT), University of Central Punjab Lahore, Pakistan
[*]Corresponding Author: Atif Raza Zaidi. Email: atif@atifraza.com

Received: October 05, 2020 **Accepted:** December 12, 2020 **Published:** March 15, 2021

Abstract: As luck would have its recent innovations in computer vision grant us to make considerable pace abate the data classification and analysis of enormous data in different organizations like training material, policy guides, and project documents that can be used internally. In addition, cloud service providers are rising in text detection techniques and offer many computer vision contributions included Google Vision, A WS Textract, and Azure OCR. In this paper image classification is achieved by CNN algorithm which is the top choice for image classification. Then using CNN, we extract text data from images. CNN has better performance rate on large datasets as it controls the problem of overfitting. Hence, the accuracy of these algorithms can be enhanced by increasing the epochs and integrating a large dataset. This approach used a methodology to resolve the number of convolution and pooling layers with the number of nodes in network. Lastly, CNN algorithms are better when data is appropriate if it has mortal features that can be analyzed and utilized by algorithm.

Keywords: Image classification; Text extraction; CNN architecture

1. Introduction

Image classification is a technique of extracting useful information from raster image which can be helpful to grasp the contents of the image. Text extraction is a technique in which we convert scanned page or classified image which have captions to ASCII code that a machine can recognize or categorize them accordingly. This technique plays a significant role in information retrieval system, digital libraries, and multimedia projects. This technique plays an important role as it can be precisely displayed the image content in the form of text to the user or it can be provided to an optical character reader for identification [1].

A multiband or raster image consists of line drawings, graphics, pixels, and sketches which are developed by historical document image data, journals scanning, handwritten textual images, printed documental images, printed multi-color books cover and newspaper imagery data. Therefore, there are many challenges in extractions techniques like data can be in low resolution, color bleeding, unknown text color and image with complex background [2].

Some commonly used terms in text features are their size i.e. the length of font size variations, and alignment is that scenes text often aligned in different directions with distorted geometric, color is that characters with same or different, edge is in strong edges at boundaries and background and distortion due to different angles of camera rotation which affect extractions accuracy [3].

Image classification can be done using many supervised algorithmic techniques included Naive Bayes, Support Vector Machine (SVM), K-Nearest Neighbor, Decision Tree, and Convolution Neural Network (CNN). Luckily these algorithmic techniques process images and classify them into many categories [4].

However, these algorithms have some problems in image classification and text data extraction. Naive Bayes regarded that the features are independent whereas in real life they depend on each other. Support Vector Machine (SVM) takes long time to train a large dataset, so it is difficult to derive a final model and variable weights, K¬Nearest Neighbor (KNN) is a slow algorithm with outlier sensitivity and curse of dimensionality, Decision Tree does not tend to work accurately due to extremely small dataset and uncertain classification.

Hence, these problems can be resolved by using Convolution Neural Network (CNN) which is remarkable algorithmic technique of Deep learning. In this paper, we have used CNN algorithm which is the best choice for image classification and text extraction from large datasets.

2. LITERATURE REVIEW

Images plays a vital role to convey messages through text and drawing in our life. To read the text given in images and classify the images of same type is the key feature in machine learning. There are different algorithms used to perform such type of tasks like CNN and RNN. CNN and RNN are appropriate for large datasets collected from different organizations.

2.1. CNN

2.1.1. LeNet

It is the first approach used m implementing Convolution Neural Network. In 1990 [5], proposed a solution for handwritten digital recognition. It is beneficial due to minimal preprocessing for large dataset. In 1998 [6], proposed a gradient based approach. They used "ConvNets" to recognize handwritten house numbers, zip codes and digits etc. As we know CNN requires computational machines with high power, but at that the machines are not able to perform such computations at high speed. It takes lot of time to perform such computations.

2.1.2. AlexNet

AlexNet is somewhat as LeNet. The main difference is of Conv and pooling layers. In AlexNet all layers are together but in LeNet Conv and pooling layers are alternate to each other. It is deeper as compared to LeNet. In 2012 [7], Krizhevsky won the competition using CNN.

2.1.3. GoogleNet

It consists of 22 layers of Inception modules but with much smaller number of parameters compared to AlexNet. In 2014 [8] developed architecture called Inception (v1). As time passes it takes many improvements. In [9], improved the Inception (v2) by introducing batch normalization. After moderation it is called Inception (v3) [10].

2.1.4. VGGNet

In 2014, Zisserman and Simonyan introduced VGGNet by studying ConvNet in depth. They extracted large parameters from network. It consists of 3x3 convolution filters.

2.1.5. Inception v4:

In 2015 [11], introduced extended version of GoogleNet. It accelerates the training process [12] of Inception Modules with residual connections [8].

2.1.6. ResNet

Introduced in 2015 [12] by Kamming et al., 1t is "Residual learning framework". In which layers learn residual functions take from inputs and prevents unreferenced functions.

3. IMAGE CLASSIFICATION AND TEXT EXTRACTION TECHNIQUES

The technique implemented in this paper broaden the perception of deep learning networks. It provides dynamic computational models along with nonlinear processing elements arranged in layers. As we know that, deep learning serves with multiple neural network techniques, here is popular approach we used: Convolution Neural Network (CNN) for image classification and text extraction.

3.1. Convolution Neural Network (CNN) and Recurrent Neural Network

Convolution Neural Network (CNN) basically assembled for image processing, but these networks have been productively used for image classification. In addition, a great success has been achieved in relevant field i.e., attribute classification, object detection, actions classification, scenes recognition, and many more.

The main convolution layer in a network associate to a subset of input commonly of size 3*3, then the next convolution layer associate to only subset of its previous layer. Thus, these layers sometimes known as feature maps can be heaped to add different filters over the input. Convolution Neural Network (CNN) also provide a feature known as pooling to reduce computational complexity; it decreases the size of output from one stack to the next in network [13].

CCN (Convolution Neural Network) and RNN (Recurrent Neural Networks) are the mainstreams in Deep Learning where CNN tackles with image classification and more extensively computer vision and RNN deals with data which has materialistic characteristics and context dependent. CNN works by compressing an image to its basic features and then by taking mixed probabilities of recognized attributes come together to determine about classification and one more advantage of CNN is that it always requires less supervision over the other classification, whereas RNN is used to examine the prediction about sequential input data i.e. videos, speech, text, sentiment analysis and machine translation, it determines the sequential features of an input image and then in a feedback loop an output is returned to the scanning pace.

4. PROPOSED METHOD

4.1. Text Convolution Neural Network

In this section, author decreased the number of filters from 128 to 120. It is examined and proved that there is no change in accuracy of model but reduces the number of parameters in an optimized network and later tested with filter size which replaced the filter size from 3 to 2 and 4 to 3 to decrease the number of parameters, hence 3*3 size filter is proved an efficient approach in many neural networks.

Over fitting is a common problem in neural networks. To avoid over fitting we implemented the L2 regularization approach as in neural networks the learning rate of the gradient decent heavily influence the performance rate. In reverse by keeping the learning rate the same during the training process, we proved learning rate degenerate for better accuracy.

4.2. Lightweight Convolution

First, we decreased the number of filters from 128 to 120. It is examined and proved that there is no change in accuracy of model but reduces the number of parameters in an optimized network and later tested with filter size which replaced the filter size from 3 to 2 and 4 to 3 to decrease the number of parameters, hence 3*3 size filter is proved an efficient approach in many neural networks.

To resolve the more memory utilization problems Lightweight CNN is a best approach. This approach works with three combinations of separable convolution, dilated convolution, and batch convolution for the less utilization of memory. Figure l shows that how CNN works on imagery data for text extraction.

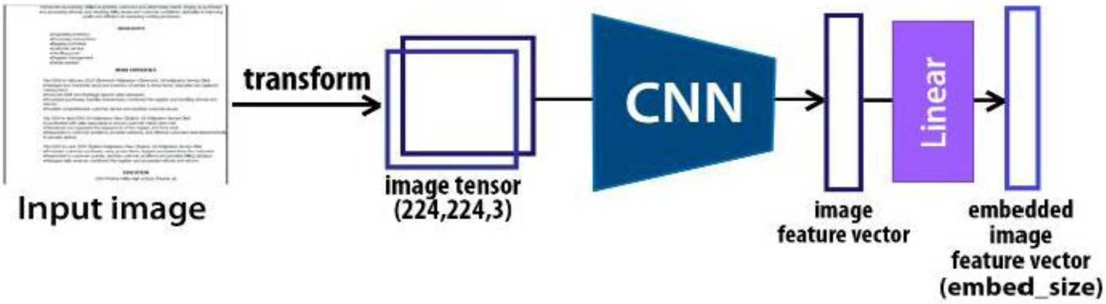

Figure 1. Working of CNN on imagery data for text extraction

4.2.1. Separable Convolution

Independent convolution is a compression method used for computational cost and memory utilization. For each channel of an input a spatial convolution implemented separately which is succeeded by a pointwise convolution like1*1, estimating the output channels depth wise convolution over a new channel. Generally, the number of channels of the input image is equal to the number of filters in convolution layer.

For text classification embedding is used which is 1D (one dimension), therefore the number of channels is one. Thus, the whole hypothesis of depthwise is not needed. It can only be beneficial to merge with the modified CNN framework.

Therefore, we use the optimized number and filter size generally as 2, 3 and 5 as described in architecture. Rather than applying 120 filters we have applied for each size 2, 3 and 5(3, 3) over a filter of each size. Then convolution is followed by pointwise convolution as (120, 1*1) filters are applied on each output separately from the last layer. 1*1 has two functions one is to expand the depth of the architecture and the second proposes nonlinearity to the architecture when it followed by ReLU.

4.2.2. Dilated Convolution:

We have used dilated convolution to decrease the number of parameters. As it is fact that 3*3 kernel will cover same field as 5*5 with rate 2. For real time segmentation dilated convolutions are suitable as it is used whenever required a deep field perspective but unable to control the cause of increment in parameters due to various convolution filters. For the presented framework we have used 3*3 dilated kernel and separable convolution filter to increase the depth of network which reduce the number of parameters.

4.2.3. Batch Convolution

For regularization purpose batch normalization layers were integrated to the network. It is used in stimulating the process of training by decreasing the covariate change.

4.2.4. Leaky ReLU

Rather than ReLU we have used leaky ReLU in the architecture as it gives nonlinearity to the network and terminate the issue of dying ReLU.

4.3. Two Optimizers

In the proposed architecture two optimizers are used i.e. Adam and SGD with their special advantages. Adam optimizer coordinates the network to learn in an efficient way and make it fast. Thus, after few epochs to make network in a stable from there’s need to slow down the learning procedure. In this way network is trained after some epochs on Adam optimizer then shifted to SGD with momentum.

5. DATASET

Dataset use for any purpose has its own importance as it reflects the real-life problems. Through dataset we can implement our algorithm and represent how well our algorithm solves the problem. As we are applying “Text Classification and Extraction”, we have taken document dataset according to our needs. We have used RVLCDIP (Ryerson Vision Lab Complex Document Information Processing). It is publicly available it has 400,000 different types of document grey-scale images. We have used 3482 images from 10 different classes (Advertisement, Form, Email, Letter, Note, Scientific, Report, Resume, News, and Memo). Dataset is complex and good example of real-life problems. We are focusing on solution to problem from simple to complex. We have picked 3 independent categories for training purpose and named it as small dataset.

6. EXPERIMENT PREPARATION

In this discuss the experiment preparation, performed the architecture in anaconda using Python language.

6.1. Data Preparation

The dataset contains 3482 images with '.tif' extension [14]. To classify the text used tesseract tool. It is used to recognize text from images. It is basically software consists of OCR "Optical Character Recognition". It takes input as image and give output in text form.

After extracting text from images, we have to prepare for testing, training and validation. For testing and training purpose make a set of data of ratio 9:1. Then the data is preprocessed removing extra things like spaces, punctuations etc.

6.2. Implementation and testing

The embedding layer is the first layer. Low Dimensional vector representations has been done through mapping vocabulary words. A lookup table has been used through learning of data. We can also use pre-trained indications.

Convolutions over the embedded word vectors has been used on the next layer by using multiple filter sizes. From this, we get three architectures having different properties. On the very 1st architecture, the base text CNN has been implemented with two hundred embedding dimensions. Each for size 5, 4 and 3, 128 filters have been used. 20 epochs have been trained through the network. In the 2nd architecture, we have been implementing the Lightweight CNN architecture mentioned above. First, the batch normalization layer has been used to get the output from the embedding layer and it is normalized.

On this Normalized output, convolution is applied. To do this, 1 filter of each of sizes two, three, five (3, 3) is applied. Convolution is applied on top of this Pointwise. It is done through convolution filter by applying 128, 1x1 filters. To accelerate training process, on the convolution layer output, batch normalization is applied to normalize it. After the mentioned operations in different architectures, RELU or leaky RELU to add non-linearity are followed in these networks, the most important feature is extracted from each output through the max pool. The fully connected layer is added to concatenate the results. For probabilistic distribution, SoftMax is applied. For the network's regularization, dropout is used.

Our goal is to reach global minima or minimum cost during the learning phase of the network. Learning rate can be reduced by the help of a decaying coefficient by the training epochs. In the last or 3rd architecture, the optimizers are modified to get better results. Two optimizers are used. Adam optimizer was the first optimizer used; it trains fast. After a few epochs, the optimizer is switched to SGD with momentum. These dual optimizer method helps to decide the correct optimizer switching condition. The accuracy graph was analyzed, we also tried multiple attempts to find the right switching condition. The switching time was depending on model, dataset, and other factors.

7. RESULTS AND EVALUATION

Comparison of networks are done according to the trainable parameters, training time, obtained loss and accuracy. Following procedure is done while comparison. Trainable parameters are recoded in the initial stage. After every 100 steps evaluation is done. Final evaluation is done at the end of training. Training time is recorded at the beginning and end of the training. Hence completion time is also measured. T-testing method is used to check the significance of the results and 0.05 is to reject the null hypothesis.

After evaluation, we got that our architecture is more compressed to the base Text CNN. With the model, we got accuracy close to the base Architecture with significantly less loss. The maximum accuracy is got by the normalization of batch. As of dual optimizers gave clearer and better results. It also improved and gave results close to the base text CNN. We used separate convolution and dilated convolution; it can give much more optimized results when used with other detailed methods in deep learning concepts. Hence, trained parameters have shown in table.

Table 1. Results of train parameters.

Method used	Dataset			
	Trainable Parameters	Accuracy	Loss	Training Time
Base Text CNN	16801044	44.6	3.20	110 hour
Optimized Text CNN	16733111	45.9	3.10	110 hour
Text CNN	16496900	41.9	1.80	59 hour
Text CNN with dual Optimizer	16496900	43.6	1.91	27 min

8. CONCLUSIONS

We have explained the base Text CNN methods and the problems associated with it. To reduce overall memory consumption, we experimented multiple optimization method on existing network. By doing different attempts, we have proposed an optimization method in this paper. All the system has been well explained along with their results. Dual optimizers gave clearer and better results. We also tried other deep learning concepts to build 3 layered architectures explained above.

From our experiments, we have proposed a solution that reduction of the number of trainable parameters by approximately 300000 happened. Along with the memory consumption reduction we achieved other benefits as well. First, loss is reduced by significant amount. Secondly, we got that training time is reduced by one fourth times.

REFERENCES

1. D. S. G. Satish Kumar, Sunil Kumar, "Text Extraction From Images," Int. J Adv. Res. Comput. Eng. Technol., 2012.
2. D. Ghai and N. Jain, "Text Extraction from Document Images-A Review," Int. J Comput. Appl., vol. 84, no. 3, 2013.
3. H. Zhang, K. Zhao, Y.-Z. Song, and J. Guo, "Text extraction from natural scene image: A survey," Neurocomputing, vol. 122,pp. 310-323,2013.
4. K. Kowsari, D. E. Brown, M. Heidarysafa, K. J. Meimandi, M. S. Gerber, and L. E. Barnes, "Hdltex: Hierarchical deep learning for text classification," in 2017 16th IEEE international conference on machine learning and applications (ICMLA), 2017, pp. 364-371.
5. LeCun et al., "Handwritten digit recognition with a back-propagation network," in Advances in neural information processing systems, 1990, pp. 396-404.
6. Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” Proc. IEEE, vol. 86, no. 11, pp. 2278–2324, 1998.
7. A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in Advances in neural information processing systems, 2012, pp. 1097–1105.
8. C. Szegedy et al., “Going deeper with convolutions,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 1–9.
9. S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” arXiv Prepr. arXiv1502.03167, 2015.
10. C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, “Rethinking the inception architecture for computer vision,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 2818–2826.
11. C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, “Inception-v4, inception-resnet and the impact of residual connections on learning,” in Thirty-first AAAI conference on artificial intelligence, 2017.
12. G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, “Improving neural networks by preventing co-adaptation of feature detectors,” arXiv Prepr. arXiv1207.0580, 2012.

13. Y. Yang, “Convolutional neural networks with recurrent neural filters,” arXiv Prepr. arXiv1808.09315, 2018.
14. RituYadav92, “Lightweighted-CNN-for-Document-Classification,” GitHub, 2020. [Online]. Available: https://github.com/RituYadav92/Lightweighted-CNN-for-Document-Classification.