An Overview of Convolutional Neural Network: Its Architecture and Applications

Shadman Sakib[1*], Nazib Ahmed[1#], Ahmed Jawad Kabir[2@], and Hridon Ahmed[1$]

[1]*Dept. of EEE, International University of Business Agriculture and Technology, Dhaka 1230, Bangladesh*
[2]Dept. of EEE, Independent University of Bangladesh, Bangladesh
sakibshadman15@gmail.com and 15305026@iubat.edu[*],ahmed.nazib45@gmail.com[#], jawadkabir192@gmail.com[@],
hridon007@gmail.com[$]
corresponding author: 15305026@iubat.edu[*]

Abstract— **With the increase of the Artificial Neural Network (ANN), machine learning has taken a forceful twist in recent times [1]. One of the most spectacular kinds of ANN design is the Convolutional Neural Network (CNN). The Convolutional Neural Network (CNN) is a technology that mixes artificial neural networks and up to date deep learning strategies. In deep learning, Convolutional Neural Network is at the center of spectacular advances. This artificial neural network has been applied to several image recognition tasks for decades [2] and attracted the eye of the researchers of the many countries in recent years as the CNN has shown promising performances in several computer vision and machine learning tasks. This paper describes the underlying architecture and various applications of Convolutional Neural Network.**

Keywords—Convolutional Neural Network (CNN), Deep learning, Architecture, Applications.

I. INTRODUCTION

A convolutional neural network is typically composed of a collection of layers which will be sorted by their functionalities. CNN especially is galvanized by the structure of the cortical area. The structural model supported a mammal's cortical region. The cortex has small areas of cells that are sensitive to the particular areas of the field of vision. This concept was enlarged by a fascinating experiment done by Hubel and Wiesel in 1962 [3]. In 1980, Neocognitron planned by Fukushima that the primary network that relies on the local connection type and graded organization between neurons for the transformation of the image. Fukushima thinks that once a bunch of parameters of the same nerve cell is affected to the same space within the entirely different position of the preorder layer, the options relative to the translational data invariableness are obtained [4]. The neocognitron in 1980 is taken into account as the precedent of ConvNets. Then LeCun et al. [5, 6] design and established the framework of CNN by developing seven learned layers including four convolutional and pooling layers followed by three fully-connected layers artificial neural network known as LeNet-5. LeNet-5 was accustomed to classify written digits and trained with the algorithmic backpropagation program [7] that made it attainable to acknowledge patterns directly from raw pixels, and therefore, eliminating a separate feature extraction mechanism. But, because of the lack of sufficient training information and computing power, this design didn't perform well under challenging issues. In 2012, Krizhevsky et al. [8] establish a CNN model that succeeded in conveyance down the error rate significantly on ILSVRC competition [9]. Krizhevsky et al. projected a deep CNN design recognized as AlexNet [8] that incontestible vital enhancement in image classification task. This architecture is thought of as a significant variant of LeNet having eight learned layers including five convolutional layers followed by three fully-connected layers. In the beginning, AlexNet is enormously equal in performances as the traditional LeNet-5 albeit it had a deeper structure than LeNet-5. But over the years, AlexNet was able to provide exceptional results compared to the previous model of ConvNets [5, 10-12] in the field of computer vision using strictly supervised learning keeping the net simple.

In this paper, we have reviewed the fundamental architecture of the convolutional neural network. Again, we have presented the development of several sophisticated CNN architectures. Finally, we have described several applications of the convolutional neural network.

II. CNN'S ARCHITECTURE

Unlike unsupervised method like Fuzzy C- Means [13] and ADBSCAN [14] clustering, CNN is a supervised method. A Convolutional Neural Network (CNN) consists of an input and an output layer, as well as multiple hidden layers. These layers are generally divided into three types: CONV, POOL, and FC (short for fully-connected). We are going to additionally expressly write the nonlinearity activation function as a layer that applies elementwise non-linearity. In this section, we tend to discuss how these layers are typically stacked along to create entire ConvNets. CNN primarily concentrate on the idea that the data contains pictures which focuses the architecture to be built in a method that most accurately fits the necessity for coping with the particular form of information. However, one of the significant variations is that the layers amidst the CNN are composed of neurons organized into three dimensions known as the spatial

dimensionality of the input (the height, the width and, the depth). The architecture of a CNN is modeled to draw advantage of the 2D structure of an input image (or various 2D input similar to a speech signal) which is often achieved with neighboring connections and tied weights followed by some pooling which causes translation invariant characteristics. A CNN consists of a range of convolutional and subsampling layers optionally accompanied through fully connected layers, i.e., one or multiple fully connected layers are present after many convolutions and pooling layers. The input and output of every stage are sets of arrays referred to as feature maps. In the case of a colored image, every feature map would be a 2D array containing a color channel of the input image, a 3D for a video and a 1D for an audio input. The output stage represents characteristics extracted from all locations on the data.
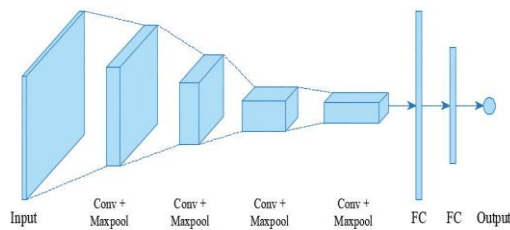
Figure 1: A basic architecture of a convolutional neural network

A. Convolution Layer

As the name implies, the convolution layer plays a significant role in how CNN operates. It forms the fundamental unit of a ConvNet wherever most of the computation is concerned. The layer's parameters focus around the use of learnable kernels. These kernels are typically tiny in spatial dimensionality, however, unfold on the whole dimension of the depth of the input. Once the information hits a convolution layer, the layer convolves every filter across the spatial dimensionality of the data to provide a 2D activation map. The output of neurons of which are connected to local regions of the input can be verified through the convolution layer through the calculation of the scalar product between their weights and also the area connected to the input volume. Neurons that consist of identical feature map shares the weight (parameter sharing) thereby reducing the complexness of the network by keeping the number of parameters low [15]. The rectified linear unit (commonly shortened to ReLU) aims to use an 'elementwise' activation function like sigmoid to the output of the activation made by the previous layer. Convolution layers can considerably scale down the complexness of the model through the optimization of its output. However, these are optimized through three hyperparameters, the depth, the stride and the setting of zero-padding. Zero-padding is the straightforward method of padding the border of the input and an efficient technique to provide additional management on the dimensionality of the

output volumes. To calculate the spatial dimensionality of the convolution layers' output, we tend to use the following formula:

$$\frac{(V - R) + 2Z}{S + 1} \ \ ...... \ (1)$$

Where V represents the input volume size, (i.e., height x breadth x depth), R represents the receptive field size, Z is the amount of zero padding set, and S refers to the stride.
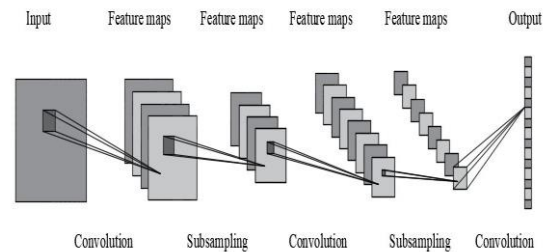
Figure 2: A CNN architecture with alternating convolution and pooling layers

The ConvNets are trained with Backpropagation algorithm, and therefore the backward pass additionally involves in convolution operation with spatially flipped filters [16]. The individual neuron within the output can represent the gradient of which might be lost across the depth, therefore solely upgrade one set of weights, as contrary to every single one.

B. Nonlinearity Layer

This layer applies the non-saturating activation function. It will increase the nonlinear properties of the choice function and of the overall network that are desirable for multi-layer networks while not affecting the receptive fields of the convolution layer. The activation functions are usually sigmoid, tanh and ReLU. Compared to different functions rectified Linear Units (ReLU) [17] are preferred as it can train neural networks many times quicker. Also, to improve the performance of the network, SOFTMAX activation function is employed at the end of the final layer.

ReLU [18]:
ReLU could be demonstrated as in eqn. (2).

$$g(y) = \begin{cases} 0 \text{ for } y < 0 \\ y \text{ for } y \geq 0 \end{cases} \ \ .... \ (2)$$

SOFTMAX [18]:
Usually, the softmax could be applied at the final layer in a convolutional neural network. Now, a softmax function could be demonstrated as in eqn. (3).

$$\alpha(c)_j = \frac{e^{c_j}}{\sum_{k=1}^{K} e^{c_k}} \quad .... \ (3)$$

C. Pooling Layer

CNN contains not solely convolution layers but also, conjointly some pooling layers. There may be a pooling layer instantly after a convolutional layer. It suggests that the outputs of the convolution layers are the inputs to the pooling layers of the network. Pooling operations cut the dimensions of the feature maps by the victimization of some functions to summarize subregions, like taking the common or the maximum value. Pooling layers aim to step by step cut the dimensionality of the data, and therefore additionally reduce the number of parameters and also the procedure complexness of the model and thus control the matter of overfitting. A number of the common pooling operations are max pooling, average pooling, stochastic pooling , spectral pooling , spatial pyramid pooling , L2-norm pooling , and multiscale orderless pooling . Figure 3 shows the operation of max pooling.
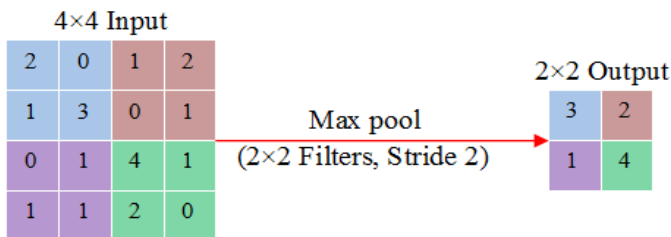
Figure 3: Max-Pooling Operation

The pooling layer operates over every activation map within the input and scales its dimensionality using the 'MAX' function. In most CNNs, these are the available shape of max-pooling layers with kernels of dimensionality of 2 x 2 applied with a stride of 2 on the spatial dimensions of the input and it scales the activation map all the way down to 25% of the original size - while maintaining the depth volume to its standard size. There are solely two usually determined strategies of max-pooling. Typically, the stride and filters of the pooling layers are each set to 2 x 2, which can permit the layer to spread through the whole area of the spatial dimensionality of the input. Moreover, overlapping pooling is also utilized, where the stride is 2 with a kernel size set to 3 x 3. However, because of the harmful nature of pooling, having a kernel size higher than three can sometimes significantly decrease the performance of the model. It is also necessary to know that apart from max-pooling, CNN architectures could contain general-pooling. General pooling layers are consists of pooling neurons which are ready to perform a large number of conventional operations simultaneously with L1/L2-normalisation, and average pooling .

D. Fully Connected Layer

The high-level reasoning within the neural network is completed via fully connected layers. In a fully connected layer Neurons at some stage have connections to all the activations in the previous layer, as seen in the conventional Multilayer Perceptron (MLP) neural networks. Their activations will thus be computed with a matrix operation followed by a bias offset.
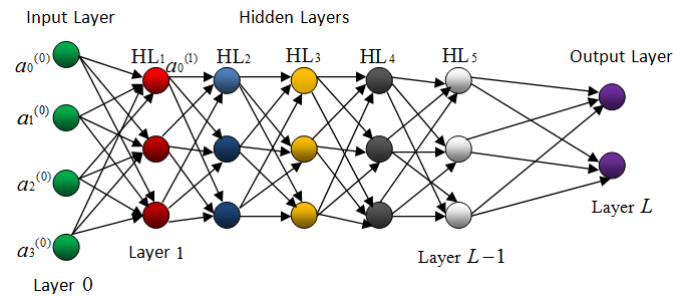
Figure 4: Fully-Connected Layers

As we can extract high-level features of the input images from the output of the convolution and pooling layers, adding a fully-connected layer is additionally an inexpensive approach to learn the non-linear combinations of these features. Though most of the features from convolutional and pooling layers could be enough for the classification task; however, combinations of those features may be even better [19]. The neurons in a fully connected layer are not spatially organized therefore there can not be a convolution layer after a fully connected layer. A fully connected layer passes the two-dimensional output to the output layer wherever we can utilize a softmax function or a sigmoid to predict the input class label. Recently, some design replaced their FC layer, as in 'Network In Network'(NIN) by Lin et al. [20], where a global average pooling layer replaces a fully connected layer. However, the objective of the fully connected layer is to flatten the high-level features, which are learned by convolutional layers, and to blend all the elements.

III. COMMON CNN ARCHITECTURES

There are many reputed CNN architectures. The description of most popular CNN architectures is given below.

A. LeNet

LeNet was the most archetype Convolutional Neural Network developed by Yann LeCun in the year of 1990 [5] and later enhanced it in 1998 [6]. The most effective best-known LeNet architecture is the one that was accustomed to read zip codes, digits, etc.

B. AlexNet

The first famous CNN architecture is AlexNet, which popularizes the convolutional neural network in Computer vision, developed by Alex Krizhevsky, Ilya Sutskever, and Geoff Hinton [8]. Later, in 2012 AlexNet was presented to the ImageNet ILSVRC challenge and considerably performed better than the second runner-up (achieved top five error rate of 16% compared to runner-up with 26% error rate). The Network had the similar architecture to LeNet; however, it was most profound, most significant architecture with all

convolution layers stacked together rather than the altering convolution and pooling layers as it was in LeNet.

C. ZF Net

After AlexNet, this convolutional neural network from Matthew Zeiler and Rob Fergus was the winner of ILSVRC 2013. It became referred to as the ZFNet (short for Zeiler & Fergus Net). It was an enhancement on AlexNet by adjusting the architecture hyperparameters, mainly by increasing the dimensions of the center convolutional layers and creating the stride and filter size on the primary layer smaller.

D. GoogleNet

This architecture developed by Szegedy et al. [21] from Google was the winner of the ILSVRC 2014 competition. It achieved a top 5 error rates of 6.67% which was very close to the human-level performance. The primary specialty of this CNN architecture was the dramatic reduction of the number of parameters within the network in its onset module known as Inception-v1 (It had only 4 million parameters compared to AlexNet with 60 million). GoogLeNet had twenty-two layers of Inception module; however, with less parameter compared to AlexNet. Later, several enhancements had been done on Inception-v1, the key being an introduction of batch normalization and RMSprop that led to Inception-v2 by Ioffe et al. [22]. There are several subsequent versions of GoogleNet with added refinements, such as Inception-v3 [23], and most presently Inception-v4.

E. VGGNet

This runner-up network in ILSVRC 2014 contest was developed by Karen Simonyan and Andrew Zisserman and referred to as VGGNet. Its main achievement was in showing that the depth of the system could be an essential factor for good performance. The final most competent network had 16 CONV/FC layers in total and, appealingly, featured an extraordinarily uniform architecture that solely performed 3x3 convolutions and 2x2 pooling from the starting to the end.

F. ResNet

Kaiming He et al. [24] developed Residual Network (ResNet). This CNN architecture features unique skip connections and essential use of batch normalization. Again, the architecture has no fully connected layers at the end of the network. The main disadvantage of this network is that it is very costly to assess because of the massive range of parameters. However, till now, ResNet is considered as state of the art Convolutional Neural Network model and is the default option for using ConvNets in practice. It had been the winner of ILSVRC 2015.

IV. APPLICATIONS OF CNN

A. Natural Language Processing (NLP)

Convolutional Neural Networks are traditionally applied in the field of Computer Vision. CNNs are usually utilized in natural language processing. CNN models are useful for numerous natural language processing issues and achieved glorious results in text categorization, semantic parsing [25], search query retrieval [26], sentence modeling [27] and classification [28], prediction [29], text categorization [30], and diversified traditional natural language processing tasks [31]. Here, we tended to explore how CNNs are utilized in text categorization and sentence classification.

1. Text Categorization

Text categorization is that the task of automatically allocating pre-defined classes to documents written in natural languages. Many kinds of text categorization are studied, each of these deals with different types of documents and classes, like topic categorization to unearth mentioned topics (e.g., sports, education, politics), spam detection [32], and sentiment classification [33-35] to notice the sentiment generally in product or movie reviews. Johnson and Zhang explained CNN on text categorization to utilize the 1D structure (namely, word order) of text data for correct prediction [36]. They directly applied CNN to high-dimensional text data, rather than low dimensional word vectors as is usually done. A typical approach to text categorization is to represent documents by bag-of-words vectors, specifically, vectors that indicate which words present in the papers, however, do not preserve word order. Word order loss which is due to bag-of-words vector is especially problematic on sentiment classification. A natural remedy is to use the word bigrams additionally to unigrams [37]. To learn from word order on text categorization, Johnson and Zhang took a unique approach, that employs convolutional neural networks [6]. They applied CNN to text categorization to form use of the 1D structure (word order) of document data so that every unit within the convolution layer corresponds to a small region of a document (a sequence of words). Gao et al. [38] propose a particular kind of deep neural network with the convolutional structure for text analysis for recommending target documents to the user depending on the material the user is reading. The system that is trained on an oversized set of web transitions, maps source-target document pairs to feature vectors, minimizing the gap between source and target documents. The work of Shen et al. [39] is of similar types. They both explain how to find out semantically purposeful representations of sentences. However, the latter outperforms the previous state of the art semantic models. Semantic embeddings from hashtags is a CNN design that predicts hashtags for Facebook posts and at a similar time generate essential embeddings for words and sentences [40]. These embeddings are then successfully applied to document recommendation task. Nowadays, there has been on-going analysis in applying CNNs on to characters [41].

2. Sentence Classification

On the practically significant task of sentence classification, Convolutional Neural Network has achieved an extraordinarily flourishing performance [27, 28, 42]. It takes the benefit of the distributed representations of words by first changing the tokens comprising every sentence into a vector, forming a matrix to be used as input. The models need not be

complicated to understand robust results: as an example, Yoon Kim planned a straightforward one-layer CNN [28] that achieved progressive (or comparable) results across many datasets. Therefore, the remarkably robust results obtained with this relatively straightforward CNN design suggest that it may function as a drop-in replacement for well-established baseline models, like Support Vector Machine (SVM) [43] or logistic regression. While further complicated deep learning models for text classification can undoubtedly continue to be developed, those deploying such technologies in practical applications can probably be interested in less complicated variants, that affords quick training and prediction times.

B. Image Recognition

Convolutional Neural Networks (CNNs) are usually employed in image recognition systems. In 2012 an error rate of 0.23 % on the MNIST database was reported [44]. In face recognition, CNNs achieved an oversized decrease in error rate [45]. CNNs were also accustomed to assessing video quality in an objective method after manual training; the ensuing system had a shallow root mean square error [46]. In the ILSVRC 2014, a large-scale visual recognition challenge, nearly all remarkably ranked team used CNN as their underlying framework. Again, the mean average precision in object detection is increased to 0.439329 and decreased classification error to 0.06656 by the winner GoogLeNet [21], which is the excellent result so far. This network had over thirty layers. The performance of convolutional neural networks on the ImageNet tests was near to that of humans. In 2015, a multi-layered CNN showed incontestable achievements to identify faces from a broad range of angles and orientations. The network trained on a database of 200,000 pictures that enclosed faces at numerous angles and directions and an additional 20 million photos without faces. They used batches of 128 images over 50,000 iterations. However, the most straightforward algorithms still struggle with little or thin objects such as a small ant on a stem of a flower or an individual holding a quill in their hand. Theses algorithms even have trouble with pictures that are distorted with filters, an increasingly common development with today's popular digital cameras.

V. CONCLUSION

In this paper, the architecture of Convolutional Neural Network (CNN) in conjunction with it's few applications, in brief, have been discussed. Also, the evolution of the various CNN architectures has been presented clearly along with their components. The CNN is better than other alternative deep learning networks in applications such as computer vision and natural language processing as it can mitigate the error rate significantly and hence improve network performances. By analyzing this paper, one can gain a better understanding of why CNN is employed in numerous applications and facilitates in several machine learning fields.

REFERENCES

[1] Mohammad Mahmudur Rahman Khan*, et al.*, "Study and Observation of the Variation of Accuracies of KNN, SVM, LMNN, ENN Algorithms on Eleven Different Datasets from UCI Machine Learning Repository," in *2018 4th International Conference on Electrical Engineering and Information & Communication Technology (iCEEiCT)*, 2018, pp. 124-129.

[2] Md. Abu Bakr Siddique*, et al.*, "Study and Observation of the Variations of Accuracies for Handwritten Digits Recognition with Various Hidden Layers and Epochs using Neural Network Algorithm," in *2018 4th International Conference on Electrical Engineering and Information & Communication Technology (iCEEiCT)*, 2018, pp. 118-123.

[3] D. H. Hubel and T. N. Wiesel, "Receptive fields, binocular interaction and functional architecture in the cat's visual cortex," *The Journal of physiology,* vol. 160, pp. 106-154, 1962.

[4] K. Fukushima and S. Miyake, "Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition," in *Competition and cooperation in neural nets*, ed: Springer, 1982, pp. 267-285.

[5] Y. 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*, et al.*, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE,* vol. 86, pp. 2278-2324, 1998.

[7] R. Hecht-Nielsen, "Theory of the backpropagation neural network," in *Neural networks for perception*, ed: Elsevier, 1992, pp. 65-93.

[8] A. Krizhevsky*, et al.*, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097-1105.

[9] A. Berg*, et al.*, "Large scale visual recognition challenge 2010," ed, 2010.

[10] K. Jarrett*, et al.*, "What is the best multi-stage architecture for object recognition?," in *Computer Vision, 2009 IEEE 12th International Conference on*, 2009, pp. 2146-2153.

[11] A. Krizhevsky and G. Hinton, "Convolutional deep belief networks on cifar-10," *Unpublished manuscript,* vol. 40, 2010.

[12] M. D. Zeiler*, et al.*, "Adaptive deconvolutional networks for mid and high level feature learning," in *Computer Vision (ICCV), 2011 IEEE International Conference on*, 2011, pp. 2018-2025.

[13] M. Siddique*, et al.*, "Implementation of Fuzzy C-Means and Possibilistic C-Means Clustering Algorithms, Cluster Tendency Analysis and Cluster Validation," *arXiv preprint arXiv:1809.08417,* 2018.

[14] Mohammad Mahmudur Rahman Khan*, et al.*, "ADBSCAN: Adaptive Density-Based Spatial Clustering of Applications with Noise for Identifying Clusters with Varying Densities," in *2018 4th International Conference on Electrical Engineering and Information & Communication Technology (iCEEiCT)*, 2018, pp. 107-111.

[15] G. E. Hinton*, et al.*, "Improving neural networks by preventing co-adaptation of feature detectors," *arXiv preprint arXiv:1207.0580,* 2012.

[16] Y. LeCun*, et al.*, "Backpropagation applied to handwritten zip code recognition," *Neural computation,* vol. 1, pp. 541-551, 1989.

[17] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th international conference on machine learning (ICML-10)*, 2010, pp. 807-814.

[18] Rezoana Bente Arif, *et al.*, "Study and Observation of the Variations of Accuracies for Handwritten Digits Recognition with Various Hidden Layers and Epochs using Convolutional Neural Network," in *2018 4th International Conference on Electrical Engineering and Information & Communication Technology (iCEEiCT)*, 2018, pp. 112-117.

[19] J. T. Springenberg, *et al.*, "Striving for simplicity: The all convolutional net," *arXiv preprint arXiv:1412.6806,* 2014.

[20] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *European conference on computer vision*, 2014, pp. 818-833.

[21] C. Szegedy, *et al.*, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1-9.

[22] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167,* 2015.

[23] C. Szegedy, *et al.*, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2818-2826.

[24] K. He, *et al.*, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770-778.

[25] E. Grefenstette, *et al.*, "A deep architecture for semantic parsing," *arXiv preprint arXiv:1404.7296,* 2014.

[26] Y. Shen, *et al.*, "Learning semantic representations using convolutional neural networks for web search," in *Proceedings of the 23rd International Conference on World Wide Web*, 2014, pp. 373-374.

[27] N. Kalchbrenner, *et al.*, "A convolutional neural network for modelling sentences," *arXiv preprint arXiv:1404.2188,* 2014.

[28] Y. Kim, "Convolutional neural networks for sentence classification," *arXiv preprint arXiv:1408.5882,* 2014.

[29] R. Collobert and J. Weston, "A unified architecture for natural language processing: Deep neural networks with multitask learning," in *Proceedings of the 25th international conference on Machine learning*, 2008, pp. 160-167.

[30] T. Mikolov, *et al.*, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781,* 2013.

[31] R. Collobert, *et al.*, "Natural language processing (almost) from scratch," *Journal of Machine Learning Research,* vol. 12, pp. 2493-2537, 2011.

[32] M. Sahami, *et al.*, "A Bayesian approach to filtering junk e-mail," in *Learning for Text Categorization: Papers from the 1998 workshop*, 1998, pp. 98-105.

[33] B. Pang, *et al.*, "Thumbs up?: sentiment classification using machine learning techniques," in *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*, 2002, pp. 79-86.

[34] B. Pang and L. Lee, "Opinion mining and sentiment analysis," *Foundations and Trends® in Information Retrieval,* vol. 2, pp. 1-135, 2008.

[35] A. L. Maas, *et al.*, "Learning word vectors for sentiment analysis," in *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies-volume 1*, 2011, pp. 142-150.

[36] R. Johnson and T. Zhang, "Effective use of word order for text categorization with convolutional neural networks," *arXiv preprint arXiv:1412.1058,* 2014.

[37] J. Blitzer, *et al.*, "Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification," in *Proceedings of the 45th annual meeting of the association of computational linguistics*, 2007, pp. 440-447.

[38] J. Gao, *et al.*, "Modeling interestingness with deep neural networks," ed: Google Patents, 2017.

[39] Y. Shen, *et al.*, "A latent semantic model with convolutional-pooling structure for information retrieval," in *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, 2014, pp. 101-110.

[40] J. Weston, *et al.*, "# tagspace: Semantic embeddings from hashtags," in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, pp. 1822-1827.

[41] C. D. Santos and B. Zadrozny, "Learning character-level representations for part-of-speech tagging," in *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, 2014, pp. 1818-1826.

[42] R. Johnson and T. Zhang, "Semi-supervised convolutional neural networks for text categorization via region embedding," in *Advances in neural information processing systems*, 2015, pp. 919-927.

[43] T. Joachims, "Text categorization with support vector machines: Learning with many relevant features," in *European conference on machine learning*, 1998, pp. 137-142.

[44] D. Cireşan, *et al.*, "Multi-column deep neural networks for image classification," *arXiv preprint arXiv:1202.2745,* 2012.

[45] S. Lawrence, *et al.*, "Face recognition: A convolutional neural-network approach," *IEEE transactions on neural networks,* vol. 8, pp. 98-113, 1997.

[46] P. Le Callet, *et al.*, "A convolutional neural network approach for objective video quality assessment," *IEEE transactions on neural networks,* vol. 17, pp. 1316-1327, 2006.