RESEARCH ARTICLE

University of KURDISTAN Hewlêr

A Study of the Applications of Convolutional Neural Networks

Ahmed S. Shamsaldin[1*], Pola Fattah[2], Tarik A. Rashid[1], Nawzad K. Al-Salihi[1]

[1]Department of Computer Science and Engineering, School of Science and Engineering, University of Kurdistan Hewler, Erbil, Kurdistan Region - F.R. Iraq

[2]Software and Informatics Engineering Department, College of Engineering, Salahaddin University-Erbil, Erbil, Kurdistan Region-F.R. Iraq

***Corresponding author's email:** ahmed.saadaldin@ukh.edu.krd

Received: 06 November 2019 **Accepted:** 10 December 2019 **Available online:** 27 December 2019

ABSTRACT

At present, deep learning is widely used in a broad range of arenas. A convolutional neural networks (CNN) is becoming the star of deep learning as it gives the best and most precise results when cracking real-world problems. In this work, a brief description of the applications of CNNs in two areas will be presented: First, in computer vision, generally, that is, scene labeling, face recognition, action recognition, and image classification; Second, in natural language processing, that is, the fields of speech recognition and text classification.

Keywords: Convolutional neural networks, Natural language, Computer vision, Deep learning

1. INTRODUCTION

The convolutional neural network (CNN) is an architecture for deep learning taken from the visual system structure. It was found by Hubel and Wiesel in 1962 during their work on the cat's primary visual cortex. The cells in the cortex are sensitive to small sub-regions of the visual field called receptive fields (Hubel and Wiesel, 1962). Detecting light in the receptive fields is done by these cells. Fukushima, 1980, proposed Neocognitron, inspired from the works of Hubel and Wiesel, which is the earliest model that had a computer simulatability. This Neocognitron is counted as the prototype of CNNs, and it was grounded on the neurons' hierarchical organization for the conversion of an image. The outline of CNNs was founded by LeCun et al., 1990, and LeCun et al., 1998, by evolving an artificial neural network with a multilayer called LeNet-5. This artificial neural network was used to perform handwritten digit classification and it was trainable by the backpropagation algorithm (Hecht-Nielsen, 1988). Training with this algorithm made it feasible to recognize patterns from raw pixels. Although LeNet-5 has many advantages, it was unsuccessful when used in solving complex problems such as video classification.

Access this article online

DOI: 10.25079/ukhjse.v3n2y2019.pp31-40 **E-ISSN:** 2520-7792

CNN has been taken into a whole new level since the initiation of general-purpose graphics processing unit GPGPUs and their usage in machine learning (Steinkraus et al., 2005). More effective techniques have been designed to train CNNs using GPU computing (Bengio et al., 2007; Chellapilla et al., 2006). A new design for CNNs was presented by Krizhevsky et al. 2012, named AlexNet that showed great enhancement in image classification. This design is close to the classic LeNet-5; however, it had a deeper structure. After the success of AlexNet, more versions were developed that demonstrated performance enhancement, versions such as GoogleNet (Szegedy et al., 2015), VGGNet (Simonyan and Zisserman, 2014), and ZFNet (Zeiler and Fergus, 2014) and ResNet (He et al., 2016).

Another version of neural networks is recurrent neural networks (RNNs) that are used for natural language processing (NLP) as it simulates the ability of humans to process language (Graves et al., 2013). However, lately, CNNs have been used in solving NLP problems such as sentiment analysis, spam detection, or topic categorization. Even though it is less natural when it comes to processing such problems, it has accomplished a competitive outcome. In addition, CNNs have been used for the problems of speech recognition. Speech is an ethereal illustration of verbal words that includes hundreds of variables and usually encounters issues of overfitting when trained using fully connected feed-forward networks (LeCun and Bengio, 1995). In addition, they do not contain integrated invariance with regard to interpretations. Shift variance is obtained automatically in CNNs, and the CNN forces the extraction of local with regard to classical architecture.

In this work, the evolvement of CNNs in computer vision and NLP fields will be demonstrated. The CNN design will be presented. Then, the applications of CNNs will be explored.

2. CNN ARCHITECTURE

The architecture of the CNNs is different from the traditional multilayer perceptron (MLP). This is to guarantee a certain degree of shift and distortion invariance (LeCun and Bengio, 1995). To do so, three design ideas are merged, which are, local receptive fields, common weights, and spatial and temporal subsampling.

Several designs of CNNs have been stated in the introduction; however, in their basic components, they are very similar. In Figure 1, the architecture of a CNN is shown (LeCun et al., 1990).

CNNs consist of multiple trainable multilayer levels (LeCun et al., 1990). Feature maps are sets of arrays that represent, for each level, the input and output (LeCun et al., 1998). If the input is a colored image, every feature map will be a two-dimensional array that holds a color channel of the inputted image, for videos it is a three-dimensional array and it is a one-dimensional array for audio input. From every location in the input, features will be exported and presented as an output in the output level.

Generally, every level contains the following: First, a non-linearity layer. Second, a filter bank layer and finally, a feature pooling layer. After several convolution and pooling layers, single or multiple fully connected layers will be present.

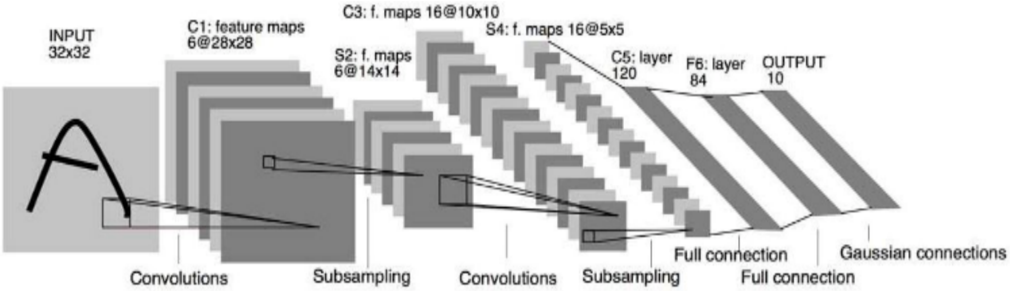

Figure 1. CNN architecture (Bhandare et al., 2016)

2.1. Convolution Layer

The core layer of CNNs is the convolutional layer. The parameters of the layer consist of learnable kernels or filters that spread through the input's full depth. The preceding layer consists of a set of elements grouped in a tiny neighborhood that send inputs to this layer. In the preceding layer, this neighborhood is known as the neuron's receptive field. When moving through each filter, the value is convolved with an input; this procedure will generate a map. When many feature maps that are produced by multiple filters get stacked together, they formulate the output of the convolution layer. The model complexity is minimized by sharing the weight vector that produces the feature map.

2.2. Non-linearity Layer

In this layer, different function layers are applied. The objective of these functions is that they should present non-linearities. These nonlinearities are required for multi-layer networks. The standard activation functions are Rectified Linear Units (ReLU), sigmoid, and tanh. However, the (ReLU) are more desirable owing to the fact that neural networks get trained several times faster (Nair and Hinton, 2010).

2.3. Pooling Layer

The convolutional layer is mostly followed by the pooling layer. This layer takes small rectangular blocks from the preceding convolutional layer and samples them to generate a single maximum output from the block (Boureau et al., 2010; Ranzato et al., 2007; Yang et al., 2009). The spatial size is minimized by the pooling layer; therefore, the parameters will also be minimized for computational purposes. In addition, this layer governs the overfitting process.

2.4. Fully Connected Layer

The high-level reasoning is conducted by one or more fully connected layers. The high-level reasoning is done by taking all the neurons in the preceding layer and linking them to every neuron in the present layer to produce global semantic information.

3. APPLICATIONS OF CNNs

In this paper, two of the main applications of CNNs will be discussed. These applications are natural language processing and computer vision.

3.1. Natural Language Processing

From the definition, extracting information from signals is one of the uses of CNNs (LeCun et al., 1990; LeCun et al., 1998). Essentially, speech is a series of signals and in NLP, one of the significant duties is to recognize it. In addition, lately, CNNs have been deployed to sentence classification, topic categorization, sentiment analysis, and many other tasks.

A. Speech Recognition: Recently, CNNs began to be implemented for speech recognition purposes, and it has shown better performance and results than deep neural networks (DNNs). In 2015, researchers in Microsoft Corporation stated four areas where CNNs are better than DNNs:

1. The robustness of the noise.
2. Distant speech recognition.
3. Low-footprint models.
4. Channel-mismatched training-test conditions (Huang et al., 2015).

When the researchers applied a CNN on 1000 hours of Kinect distance, they obtained a 4% Word Error Rate Reduction (WERR) compared with a DNN on a similar size. Kinect is a series of motion sensors developed by Microsoft that enables users to interact with their computers using signs and voice commands (Zhang, 2012). Kinect distance is the distance that the device supports which is 1.2 to 3.5 m (Zhang, 2012). The CNN structure, with maxout units, has been used for implementing small-footprint models to devices to get 9.3% WERR from DNNs. To increase the robustness of CNNs, the polling needs to be done at a local frequency region. To avoid over-fitting, less parameters are used to extract the low-level features. Palaz and Collobert, 2015, state that for CNNs, implementing direct modeling is doable for the connection between raw speech signal and the phones. In addition, in comparison to the classical methods, an Automatic Speech Recognition (ASR) system is implementable. In this work, it is demonstrated that such ASR systems' characteristics are impacted by noise less than the Mel Frequency Cepstral Coefficients (MFCC) characteristics.

There are two types of microphones to acquire the distant speech: Single Distant Microphone (SDM) and Multiple Distance Microphone (MDM). Although, there are several issues when dealing with distant speech, the main ones are multiple audio sources and continuous noise. Swietojanski and Arnab, 2014, discovered that, for distant speech recognition, a CNN enhances the WERR by 6.5% compared with a DNN and 15.7% over the Gaussian Mixture Model (GMM). The WERR improved by 3.5% relative to DNNs and 9.7% over the GMM, for cross-channel convolution. Commonly, RNNs are more popular in the case of Distant Speech Recognition owing to the accurate results that it produces. Stanford University researchers merged the RNN and CNN methods to obtain better results. In this combination, CNNs are deployed for frame-level classification and RNNs are implemented with a Connectionist Temporal Classification to decode the frames in a sequence of phonemes. They achieved 22.1% on a TIMIT dataset with CNNs, and the phone sequence has an error of 29.4% (Song and Cai, 2015).

Nowadays, the main application in human-centered signal processing is Speech Emotion Recognition (SER) (Mao et al., 2014). To learn the salient features of SER, CNNs are implemented. Mao et al., 2014, trained CNNs for SER in two phases. First, the sparse encoder (SAE) is used to lean the local invariant features (LIF). Second, the LIF is fed into the salient descriptive feature analysis. This system proved to be steady in complicated scenarios. Zheng et al, 2015, used labelled training audio data to train a deep CNN for SER. To overcome the interference and minimize the dimensionality, they implemented the principle component analysis (PCA) technique. There were 2 convolutional and 2 pooling layers in the system and acquired 40% accuracy for classification. Using hand-crafted audio features, the system was better than SVM-based classification.

Over the years, researchers have been developing and building systems to deal with the significant issue of unwanted noise and minimizing it. The finest pooling, padding, and input feature map selection techniques were deployed by Qian et al., 2016, and it was tested on two tasks, first, Aurora4 Task and second, AMI meeting transcription task to evaluate its robustness. The design gained 17% enhancements compared with LSTM-RNN on Aurora4 and 10% decrease compared with the standard CNN in AMI.

B. Text classification: Documents and sentences are shown as matrixes and they are dealt with using the NLP tasks. Every token represents a row in the matrix which can be a word or character. Therefore, every row is basically a vector that is a token. These vectors are low-dimensional representations named as word embeddings. Word embedding is a group of language modelling and feature learning technique in NLP in which words from the vocabulary are mapped to vectors of real numbers in a low-dimensional space (Mikolov et al., 2013).

There are word embedding methods, for example, Word2vec suggested by Mikolov et al., 2013, and GloVe by Pennington et al., 2014. Using 100 billion words from Google News that are accessible publicly, the Word2vec was trained. In addition, using a fixed or varying filter size, the convolution is computed and the feature map is produced. For every feature map, pooling is executed. A final characteristics vector is produced and run through a final layer to complete the required tasks, for example, classification. In NLP, the main features of CNNs, location invariance and local compositionality, do not apply like they do in computer vision applications. The place of a word in a sentence is extremely important. In the case of pixels, the ones that are close to each other may belong to the same object and can be connected; however, in sentences, this does not apply as words that are close to each other do not necessarily have the same meaning, and consequently, they might not be connected. Hence, CNNs are applied to do classification tasks only, tasks such as topic categorization or sentiment analysis. For classical CNNs, it is difficult to perform tasks such as PoS tagging or entry extraction owing to the fact that sequence is important in these tasks, and convolution and pooling processes do not keep track of the sequence of the words.

The CNN was trained by Johnson and Zhang, 2014, with no vector of pre-trained words. In other words, high-dimensional data are used straightforwardly. In a second strategy, in the convolutional layer, the writers used a bag-of-words conversion. Both strategies overtook the other methods through minimizing the error rate to around 2% and 1.5% correspondingly. Pennington et al., 2014, developed their model by adding an unsupervised learning which realizes embeddings of test regions. This model overtook the earlier one through enhancing the results by almost 0.9%. For lengthy text, the method mentioned performs well; however, for short text it is unconfirmed.

Different hyper parameters have to be taken in consideration when developing CNN designs such as the input representations, for example, Word2vec or one-hot, pooling methods, activation functions, and filters. To discover the effect of the hyper parameters mentioned earlier in the model functioning, Johnson and Zhang, 2015, executed a sensitivity analysis through changing different hyper parameters. They concluded that non-static word embeddings are better to be avoided when handling big datasets; the performance is directly affected by the filter size, and pooling is enhanced when using the 1-max pooling method.

Recently, there has been progress in research studies to implement CNNs to characters directly. Santos and Zadrozny, 2014, worked on learning character-level embeddings and joining them with a pre-trained word embedding. Afterward, CNNs were used for Part Speech tagging. It gave results in two languages, English with an accuracy of 97.32% on Penn Tree-bank WSJ corpus and Portuguese having 97.47% accuracy on the Mac-Morphus Corpus where the error was minimized by approximately 12% in comparison to the best previous result. In both publications (Kim et al., 2016; Zhang et al., 2015), CNNs were used to directly learn from characters. They implemented the model to sentiment analysis and text categorization with the aid of a deep network. The outcome changed depending on the dataset size, the alphabet selection, and whether the text is curated or not. Despite having 60% less parameter, it functions on par with existing state-of-the-art results on the English Penn Tree-bank. In languages with high morphology, it overtakes the preceding word-level models.

3.2. Computer Vision

To recognize the structure of an image, CNNs are deployed. Normally, an image is fed into the network as a grid of numbers; however, a better way to do it is by breaking the image into intersecting image tiles that are sent to a small neural network.

CNNs consist of multiple trainable multilayer levels (LeCun et al., 1990). Feature maps are sets of arrays that represent, for each level, the input and output (LeCun et al., 1998). If the input is a colored image, every feature map will be a two-dimensional array that holds a color channel of the inputted image, and for videos it is a three-dimensional array. At every location in the inputted image, a feature is extracted. These extracted features are shown, at the output, as a feature map. Every level contains the following, first, a non-linearity layer, second, a filter bank layer, and, finally, a feature pooling layer. A classical CNN might contain up to three of these 3-layer levels and a categorization module that follows these levels.

A. *Face Recognition*: a sequence of correlated problems arises with face recognition, these are as follows:

1. Recognizing the faces in the picture.
2. Focusing on every face even if the quality is low or the face is displayed in different poses.
3. Recognizing unique characteristics.
4. Matching the recognized characteristics with the ones in the database and specifying the person's name.

Faces are complex, multidimensional, visual stimuli that were displayed by a hybrid neural network merging local picture sampling, a self-organizing map neural network, and a CNN. Karhunen-Loe`ve Transform was used to show the results instead of the self-organizing map that performed well (5.3% error versus 3.8%) and a multilayer perceptron that did badly (40% error versus 3.8%) (Lawrence et al., 1997).

B. *Scene Labeling:* In scene labeling, every pixel is categorized under the class of the object that it fits into. Farabet et al., 2012, suggested using a multiscale CNN which resulted in highest precisions on the Shift Floe Dataset (33 classes) and the Barcelona Dataset (170 classes) and close to the highest accuracy on the Stanford Background Dataset (8 classes). Their proposed method generated a 320-by-240 image labeling in less than a second with characteristics extraction (Farabet et al., 2012).

The recurrent design for CNNs proposes a sequential series of networks that share a similar parameter set (Pinheiro and Collobert, 2014). Automatically, the network will learn to smoothen the labels that have been predicted. In addition, the system will recognize and fix the errors as the size of the context grows with the built-in recurrence. Regions with CNN characteristics (R-CNN) is a modest and scalable detection algorithm that enhances the mean average precision (mAP) by more than 30% compared with the previous best result on VOC 2012 that attained an mAP of 53.3% which was proposed by researchers at UCB and ICSI (Girshick et al., 2014).

The methods of CNNs that have been mentioned earlier were deployed for semantic segmentation. For that purpose, every pixel that was categorized under the category of its region had some problems that were tackled by the fully convolutional networks that are trained end-to-end or pixel-to-pixel. Fully convolutional networks customized from contemporary classification networks such as AlexNet (Krizhevsky et al., 2012), GoogleNet (Simonyan and Zisserman, 2014), and VGG net (Szegedy et all., 2015) accomplish the state-of-the-art segmentation of PASCAL VOC (20% relative improvement to 62.2% mean IU on 2012), NYUDv2, and SIFT Flow, though reading takes no more than one-fifth of a second for a standard image (Long et al., 2015).

In the last two years, deep convolutional neural networks (DCNNs) have enhanced the functioning of computer systems regarding the problems concerning image classification (Krizhevsky et al., 2013; Papandreou et al., 2014; Sermanet et al., 2013; Simonyan and Zisserman, 2014; Szegedy et al., 2014).

C. *Image Classification:* As CNNs have the joint feature and classifier learning ability, they produce better classification accuracy compared with the other methods when operated on large-scale datasets (Gu et al., 2018). Krizhevsky et al., 2012, developed the AlexNet and attained the best performance in ILSVRC 2012. After this success, several other works accomplished important enhancements in the accuracy of the classification through minimizing the filter size (Strigl et al., 2010) or increasing the network's depth (Simonyan and Zisserman, 2014; Szegedy et al., 2015).

A quick, completely parameterizable GPU usage of CNN, distributed benchmark outcomes for object detection (NORB, CIFAR10) with blunder rates of 2.53%, 19.51%. Lowe, 1999, shows that a GPU code for picture classification is up to two times quicker than its CPU counterpart. Strigl et al., 2010, and Uetz and Behnke, 2009, state that multicolumn deep neural networks (MCDNNs) can beat every past strategy for image classification and show that pre-preparing is not needed (though once in a while helpful for small datasets) while diminishing the mistake rate by 30%-40% (Cireşan et al., 2012). Non-saturating neurons and effective GPU execution of the convolution operation brought about a triumphant best 5 test mistake rate of 15.3%, contrasted with 26.2% accomplished by the second-best entry in the ILSVRC-2012 contest for the categorization of 1.2 million high-resolution pictures in the ImageNet LSVRC-2010 challenge into the 1000 unique classes (Krizhevsky et al., 2012).

On the basis of the fact that some classes in image classification are more ambiguous than others, the Hierarchical Deep Convolutional Neural Network (HD-CNN) was developed. It is based on the conventional CNNs that are N-way classifiers and follow the coarse-to-fine classification strategy and design module. HD-CNN with a CIFAR100-NIN building block shows a testing accuracy of 65.33% which is better than the accuracy of other standard deep models and HD-CNN models on the CIFAR100 dataset (Yan et al., 2015).

Image classification systems that deal with fine-grained images are grounded on the concept of recognizing the foreground objects to distinguish unique features. Applying attention to fine-grained categorizing can be done with the least monitoring settings at which only the class label is given. This can be done using the attention taken from the CNN that is trained with a categorizing task. This is the opposite to the other techniques that need an object bounding box or a part landmark to train or test. For the CUB200-2011 dataset, under the poorest supervision setting, this method produces the highest accuracy (Xiao et al., 2015).

D. *Action Recognition:* The trouble with creating a system for action recognition lies in the translation and spread of characteristics in various patterns that belong to the same action class. The old methods included the building of motion history, the usage of Hidden Markov Models, or the more up-to-date action sketch generation. The modified CNN model has a three-dimensional receptive field structure. This three-dimensional field structure offers translation invariant feature extraction capability. In addition, using a shared weight minimizes, in the action recognition system, the number of parameters. At Stanford University, researchers proposed a development to the standard methods used in visual recognition which were based on SIFT, proposed by Lowe, 1999, and HOG, proposed by Dalal and Triggs, 2005, by deploying the Independent Subspace Analysis (ISA) algorithm that is an extension of the Independent Component Analysis (ICA) that is famous for its use in natural image statistics (Dalal and Triggs, 2005). The ISA algorithm to learn invariant spatio-temporal characteristics from uncategorized video data was applied on the Hollywood2 and YouTube action datasets and it resulted in 53.3% and 75.8% accuracy percentage, respectively. This percentage is almost 5% better than the accuracy outcome published earlier.

Wang et al., 2016, show that the temporal pyramid pooling–based CNN that is used for action recognition overcomes the possibility of overlooking the significant frames and needs less training data and gives better outcomes when applied on Hollywood2 and HMDB51 datasets. Two stream CNN designs which merge both spatial and temporal systems deliver competitive outcomes on the standard UCF101 and HMDB51 video activity benchmarks (Simonyan and Zisserman, 2014). For recognizing human action in videos, a Pose-based Convolutional Neural Network (P-CNN) descriptor is applied (Chéron et al., 2015). Functioning of appearance-based (App) and flow-based (OF) P-CNN demonstrates an accuracy of 73.4%maP for JHMDB-GT and 60.8%maP for the MPPII - Cooking Pose estimation datasets. R*CNN (Gkioxari et al., 2015) trains action-specific models and feature maps together which accomplishes 90.2% mean AP on the Pascal VOC Action dataset overtaking the other methods in the arena by a high margin.

By implementing three-dimensional convolutions, the three-dimensional CNN model for action recognition identifies and isolates characteristics, thus catching the motion information encoded in several neighboring frames. Intelligent video surveillance, customer characteristics, and shopping behavior analysis are instances of real-world environments in which three-dimensional CNN beats the cube frame–based two-dimensional CNN model, SPM cube gray, and SPM MEHI. Ji et al., 2012, state that the three-dimensional CNN model is most efficient when there is a lower number of positive samples and attains a general accuracy of 90.2% as compared with 91.7% realized by the HMAX model (Jhuang et al., 2007). The reconfigurable CNN proposed by Wang et al, 2014, optimized the present methods and accomplished an accuracy average of 81.2% on the CAD120 dataset, 60.1% on the OA1 dataset, and 45.0% on the OA2 dataset. The accuracies obtained by reconfigurable CNNs are much better than the accuracy results demonstrated by the models proposed by Ji et al., 2012, and Xia and Aggarwal, 2013.

For higher performance, the computations need to be scaled up to enable large datasets and quicken the training on the models. Now, this is a pushing need for three-dimensional deep learning models with extended connectivity using CNNs. This can be accomplished utilizing multicore CPUs and GPUs by accomplishing data and model parallelism through making the preparation of models parallel. Rajeswar et al, 2015, determined that the three-dimensional CNN code scales up the greatest on CPUs if the convolution step is applied with a highly parallel FFT-based method, thus accomplishing the performance similar to GPUs using OpenMP.

4. CONCLUSION

As shown in this work, CNN offers better accuracy when compared with other standard approaches. In addition, it enhances the performance because of the special features it has such as shared weights and local connectivity. In applications related to computer vision and natural language processing, CNN has proven its superiority as it lessens the usual problems.

REFERENCES

Bengio, Y., Lamblin, P., Popovici, D. & Larochelle, H., (2007). Greedy layer-wise training of deep networks. In *Advances in neural information processing systems,* 153-160.

Bhandare, A., Bhide, M., Gokhale, P. & Chandavarkar, R., (2016). Applications of convolutional neural networks. *International Journal of Computer Science and Information Technologies*, *7*(5), 2206-2215.

Boureau, Y. L., Ponce, J. & LeCun, Y. (2010). A theoretical analysis of feature pooling in visual recognition. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, 111-118.

Chellapilla, K., Puri, S. & Simard, P. (2006). High performance convolutional neural networks for document processing. *Tenth International Workshop on Frontiers in Handwriting Recognition*, Université de Rennes 1, Oct 2006, La Baule (France). ⟨inria-00112631⟩

Chéron, G., Laptev, I. & Schmid, C. (2015). P-cnn: Pose-based cnn features for action recognition. In *Proceedings of the IEEE international conference on computer vision*, 3218-3226.

Cireşan, D., Meier, U. & Schmidhuber, J. (2012). Multi-column deep neural networks for image classification. *arXiv preprint arXiv:1202.2745*.

Dalal, N. & Triggs, B. (2005). Histograms of oriented gradients for human detection.

Farabet, C., Couprie, C., Najman, L. & LeCun, Y. (2012). Learning hierarchical features for scene labeling. *IEEE transactions on pattern analysis and machine intelligence*, *35*(8), 1915-1929.

Fukushima, K. (1980). Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological cybernetics*, *36*(4), 193-202.

Girshick, R., Donahue, J., Darrell, T. & Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition,* 580-587.

Gkioxari, G., Girshick, R. & Malik, J. (2015). Contextual action recognition with r* cnn. *ICCV '15 Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV),* 1080-1088. DOI: 10.1109/ICCV.2015.129.

Graves, A., Mohamed, A.R. & Hinton, G. (2013). Speech recognition with deep recurrent neural networks. In *2013 IEEE international conference on acoustics, speech and signal processing,* 6645-6649.

Gu, J., Wang, Z., Kuen, J., Ma, L., Shahroudy, A., Shuai, B., Liu, T., Wang, X., Wang, G., Cai, J. & Chen, T. (2018). Recent advances in convolutional neural networks. *Pattern Recognition*, *77*, 354-377.

He, K., Zhang, X., Ren, S. & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition,* 770-778.

Hecht-Nielsen, R. (1988). Theory of the backpropagation neural network. *Neural Networks*, 1, 445.

Hinton, G.E., Osindero, S. & Teh, Y.W. (2006). A fast learning algorithm for deep belief nets. *Neural computation*, 18(7), 1527-1554.

Huang, J.T., Li, J. & Gong, Y. (2015). An analysis of convolutional neural networks for speech recognition. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP),* 4989-4993.

Hubel, Hubel, D.H. & Wiesel, T.N. (1962). Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *The Journal of physiology*, 160(1), 106-154.

Ji, S., Xu, W., Yang, M. & Yu, K. (2012). 3D convolutional neural networks for human action recognition. *IEEE transactions on pattern analysis and machine intelligence*, 35(1), 221-231.

Johnson, R. & Zhang, T. (2014). Effective use of word order for text categorization with convolutional neural networks. *arXiv preprint arXiv:1412.1058*.

Johnson, R. & Zhang, T. (2015). Semi-supervised convolutional neural networks for text categorization via region embedding. In *Advances in neural information processing systems,* 919-927.

Kim, Y., Jernite, Y., Sontag, D. & Rush, A.M. (2016). Character-aware neural language models. In *Thirtieth AAAI Conference on Artificial Intelligence*.

Krizhevsky, A., Sutskever, I. & Hinton, G.E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems,* 1097-1105.

Lawrence, S., Giles, C.L., Tsoi, A.C. & Back, A.D. (1997). Face recognition: A convolutional neural-network approach. *IEEE transactions on neural networks*, 8(1), 98-113.

LeCun, Y., Boser, B.E., Denker, J.S., Henderson, D., Howard, R.E., Hubbard, W.E. & Jackel, L.D. (1990). Handwritten digit recognition with a back-propagation network. In *Advances in neural information processing systems,* 396-404.

LeCun, Y., Bottou, L., Bengio, Y. & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11), 2278-2324.

LeCun, Y. & Bengio, Y. (1995). Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, 3361(10), 1995.

Long, J., Shelhamer, E. & Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition,* 3431-3440.

Lowe, D.G. (1999). Object recognition from local scale-invariant features. In *iccv,* 99(2), 1150-1157.

Mao, Q., Dong, M., Huang, Z. & Zhan, Y. (2014). Learning salient features for speech emotion recognition using convolutional neural networks. *IEEE transactions on multimedia*, 16(8), 2203-2213.

Mikolov, T., Chen, K., Corrado, G. & Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Nair, V. & Hinton, G.E. (2010). Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10),* 807-814.

Palaz, D. & Collobert, R. (2015). *Analysis of cnn-based speech recognition system using raw speech as input* (No. REP_WORK). Idiap.

Pennington, J., Socher, R. & Manning, C. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP),* 1532-1543.

Pinheiro, P.H. & Collobert, R. (2014). Recurrent convolutional neural networks for scene labeling. In *31st International Conference on Machine Learning (ICML)* (No. CONF).

Qian, Y., Bi, M., Tan, T. & Yu, K. (2016). Very deep convolutional neural networks for noise robust speech recognition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 24(12), 2263-2276.

Rajeswar, M.S., Sankar, A.R., Balasubramaniam, V.N. & Sudheer, C.D. (2015). Scaling up the training of deep CNNs for human action recognition. In *2015 IEEE International Parallel and Distributed Processing Symposium Workshop,* 1172-1177.

Ranzato, M.A., Huang, F.J., Boureau, Y.L. & LeCun, Y. (2007). Unsupervised learning of invariant feature hierarchies with applications to object recognition. In *2007 IEEE conference on computer vision and pattern recognition,* 1-8.

Santos, C.D. & Zadrozny, B. (2014). Learning character-level representations for part-of-speech tagging. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14),* 1818-1826.

Shamsaldin, A., Rashid, T., Al-Rashid Agha, R., Al-Salihi, N. and Mohammadi, M. (2019). Donkey and smuggler optimization algorithm: A collaborative working approach to path finding. *Journal of Computational Design and Engineering*, 6(4), pp.562-583.

https://doi.org/10.1016/j.jcde.2019.04.004

Simonyan, K. & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Simonyan, K. & Zisserman, A. (2014). Two-stream convolutional networks for action recognition in videos. In *Advances in neural information processing systems,* 568-576.

Song, W. & Cai, J. (2015). End-to-end deep neural network for automatic speech recognition. *Standford CS224D Reports*.

Steinkraus, D., Buck, I. & Simard, P.Y. (2005). Using GPUs for machine learning algorithms. In *Eighth International Conference on Document Analysis and Recognition (ICDAR'05),* 1115-1120.

Strigl, D., Kofler, K. & Podlipnig, S. (2010). Performance and scalability of GPU-based convolutional neural networks. In *2010 18th Euromicro Conference on Parallel, Distributed and Network-based Processing,* 317-324.

Swietojanski, P. & Arnab G. (2014). Convolutional neural networks for recognition. *IEEE Signal Processing Letters* 21.9 (2014), 1120-1124.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V. & Rabinovich, A. (2015). Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition,* 1-9.

Uetz, R. & Behnke, S. (2009). Large-scale object recognition with CUDA-accelerated hierarchical neural networks. In *2009 IEEE international conference on intelligent computing and intelligent systems,* 1, 536-541.

Wang, K., Wang, X., Lin, L., Wang, M. & Zuo, W. (2014). 3d human activity recognition with reconfigurable convolutional neural networks. In *Proceedings of the 22nd ACM international conference on Multimedia,* 97-106.

Wang, P., Cao, Y., Shen, C., Liu, L. & Shen, H.T. (2016). Temporal pyramid pooling-based convolutional neural network for action recognition. *IEEE Transactions on Circuits and Systems for Video Technology*, 27(12), 2613-2622.

Xia, L. & Aggarwal, J. K. (2013). Spatio-temporal depth cuboid similarity feature for activity recognition using depth camera. In *Proceedings of the IEEE conference on computer vision and pattern recognition,* 2834-2841.

Xiao, T., Xu, Y., Yang, K., Zhang, J., Peng, Y. & Zhang, Z. (2015). The application of two-level attention models in deep convolutional neural network for fine-grained image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,* 842-850.

Yan, Z., Zhang, H., Piramuthu, R., Jagadeesh, V., DeCoste, D., Di, W. & Yu, Y. (2015). HD-CNN: hierarchical deep convolutional neural networks for large scale visual recognition. In *Proceedings of the IEEE international conference on computer vision,* 2740-2748.

Yang, J., Yu, K., Gong, Y. & Huang, T. (2009). Linear spatial pyramid matching using sparse coding for image classification. In *2009 IEEE Conference on computer vision and pattern recognition,* 1794-1801.

Zeiler, M.D. & Fergus, R. (2014). Visualizing and understanding convolutional networks. In *European conference on computer vision.* Springer, Cham.*,* 818-833.

Zhang, Z. (2012). Microsoft kinect sensor and its effect. *IEEE multimedia*, 19(2), 4-10.

Zhang, X., Zhao, J. & LeCun, Y. (2015). Character-level convolutional networks for text classification. In *Advances in neural information processing systems,* 649-657.

Zheng, W.Q., Yu, J.S. & Zou, Y.X. (2015). An experimental study of speech emotion recognition based on deep convolutional neural networks. In *2015 international conference on affective computing and intelligent interaction (ACII),* 827-831.