Convolutional Neural Network and its Applications in Artificial Intelligence

Dr. Negm Eldin Mohamed Shawky
High Institute for Computers and
Information Systems
Al- Shorouk Academy
Cairo – Egypt
e.mail: negmshawky@gmail.com

Abstract

Deep learning is an artificial intelligence (AI) function that specially based on Convolutional neural networks (CNNs) techniques that simulate the workings of human brain for processing data and are able to learn without human supervision. This techniques are widely used in many applications of pattern and image recognition with obtaining a number of advantages compared to other traditional techniques. This study covers the basics of CNNs including a description to the functions of its various layers and mathematically learns how to use it in many AI applications. The structure models of these applications that are based on CNNs are proposed to show the improvement in processing time and state-of-the-art precision. This study also provides an overview of different CNNs architectures that are used in famous useful applications for different fields in our life based on computer vision tasks.

Key Words: ***Convolutional Neural Networks (CNNs), artificial intelligence (AI), deep learning, image recognition, speech recognition, biometrics.***

1. INTRODUCTION

Convolutional Neural Networks (CNNs) are considered as structure of Deep Learning which are based on a neural network technique.

The neural network is a system of artificial "neurons" that are connected to exchange messages between them. The connections are assigned with numeric weights that are updated during the training process, so that the correct training process when reach to optimum values can be used in image or pattern recognition. The network consists of multiple layers with many neurons. Each layer responds to different combinations of inputs from the previous layers. According to the structure of layers the first layer detects a set of prehistoric patterns in the input; all the input of next layers depend on the output of previous layers.

As mentioned before the CNN model is considered as neural network with a deep structure. This structure is stimulated by the human brain system to learn abstract concepts from information being processed by sensory margin as in object detection and pattern classification. Also CNN model has a huge success in the fields of computer vision and artificial intelligence. It has been organized in many applications such as traffic sign detection and identification [1, 2], indoor object detection and recognition [3-5] and many other applications: Learning language, understanding speech and recognizing faces, [6, 7].

Using CNN, automatic pattern recognition and its applications has been an active field of AI research, aiming to generate machines that communicate with people via face, speech, finger, hand, etc. [8]. Recently CNN can be used in application to help the children who are not able to understand the emotional states of the speakers to develop poor social skills [9, 10].

The best possible correct detection rates (CDRs) have been considered for measuring performance in different applications using CNNs [11-14]. CNNs not only give the best performance compared to other detection algorithms, but also outperform humans in cases such as classifying objects into fine-grained categories [14].

In this study, investigate the structure of CNNs that consist of many connected layers with different function for each layer will be discussed to perform how these layers operate. CNNs models and its capabilities that have been used in major different fields of AI applications will be described. An overview of many applications in different fields as in image or object detection, satellite image analysis, face and speech recognition with classifying the emotions of people who interact with the automated machine using pre created databases within and across different states, understanding climate, advertising, optical character recognition, barcode/QR code recognition and any other interesting filed will be illustrated to show the importance of using CNNs.

2. CONVOLUTIONAL NEURAL NETWORKS(CNNS)

As mentioned above, CNNs are one of the most popular deep learning models that have clear remarkable success in different research areas of AI such as object recognition [15], face recognition [16], handwriting recognition [17], speech recognition [18], and natural language processing [19]. The convolution name comes from the fact that convolution refer to the mathematical operation in these neural networks. Generally, CNNs have four basic building blocks as will be described below: the convolutional layer, the pooling layer, the activation layer, and the fully connected layer. The building blocks will be described along with some basic concepts such as SoftMax unit, Rectified linear Unit, and Dropout. According to the architecture of the CNN as shown in figure 1, the convolution layer (CONV) which processes the received input data. The pooling layer (POOL) provides compressing of the information (often from pooling) by reducing the size of the intermediate image. The activation layer (Rectified linear Unit) often called as 'ReLU' which refer to the activation function that is used in this layer. The “Fully Connected” (FC) layer which is considered as a Multiple-perceptron layer (MLP) of neural network algorithms. The classification layer (SoftMax) that predicts the class of the input image.

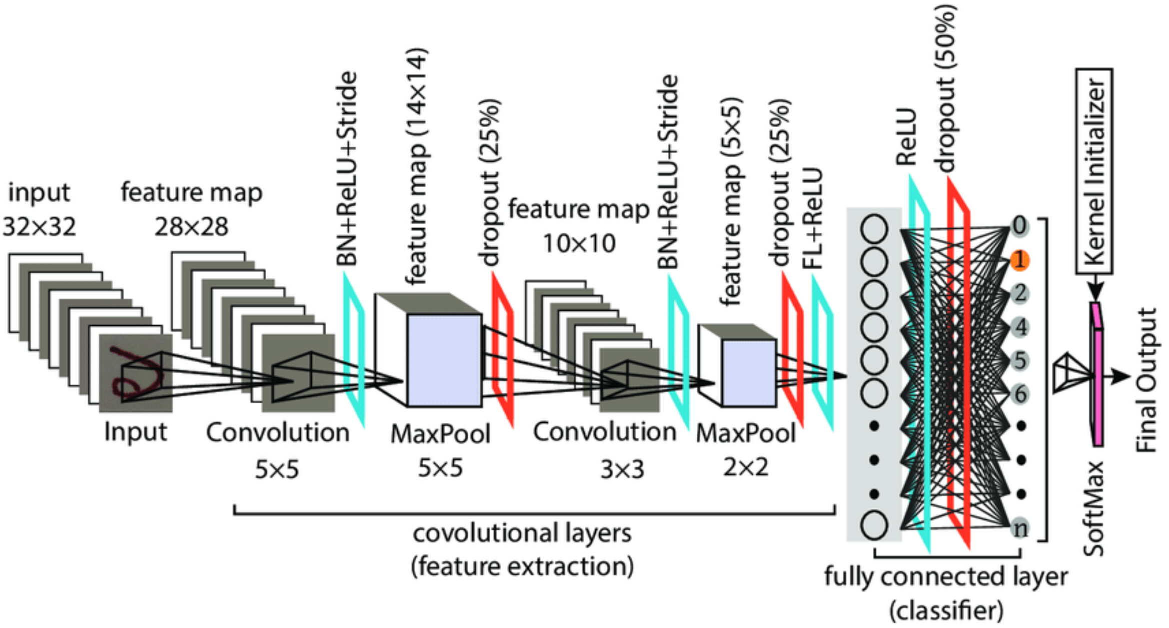


Figure 1: The architecture of the CNN used in different applications

2.1 Convolution layers

According to feature extraction process used in image recognition which is similar to the operation of convolution that extracts different features of the input image. The first convolution layer extracts low-level features like edges, lines, and corners while the next layers of convolution extract higher-level features. Figure 2 illustrates the process of 3D convolution used in CNN. The input image is of size N x N x D and is convolved with H kernels, each of size k x k x D separately. Convolution of an input image with one kernel produces one output feature, and with H kernels independently produces H features. Starting from top-left corner of the input image, each kernel size is moved from left to right, one element at a time. In the first, once the top-right corner is reached, the kernel is moved one element in a downward direction, and this movement is repeated again where the kernel is moved from left to right one element at a time. Finally this process is repeated until the kernel reaches the bottom-right corner. For example as shown in figure 3, For the case when N = 7 and k = 3 which need to be convolved to obtain the output as a result of N*k, kernel can take 5 unique positions from left to right and 5 unique positions from top to bottom. Corresponding to these positions, each feature in the output will contain 5x5 (i.e., (N-k+1) x (N-k+1)) elements. For each position of the kernel in a sliding window process, k x k x D elements of input (part of input N x N x D) and k x k x D elements of kernel are element-by element multiplied and accumulated. So to create one element of one output feature (one element of N* k), k x k x D of both input and kernel are convolved where multiply-accumulate operations are required.

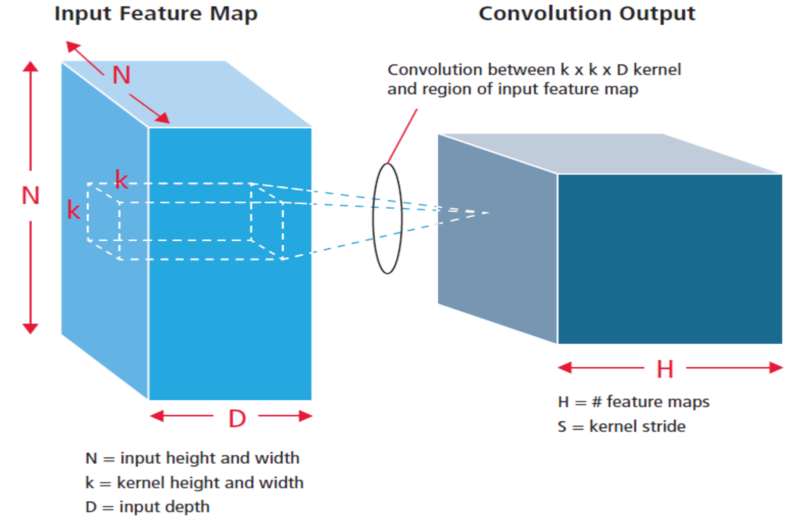
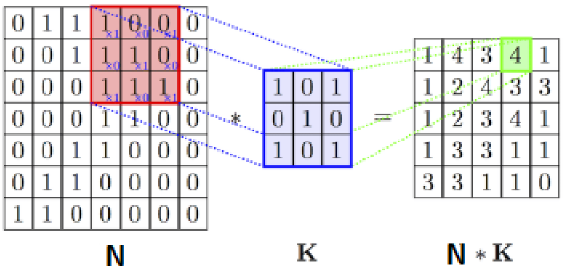

Input Feature Map
Convolution Output
N
N
k
k
D
Convolution between k x k x D kernel and region of input feature map
H
H = # feature maps
S = kernel stride
N = input height and width
k = kernel height and width
D = input depth

Figure 2: graphical representation of convolution process

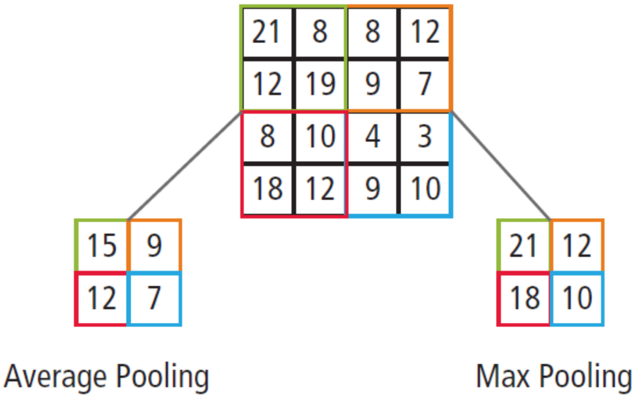


$$\begin{bmatrix} 0&1&1&1&0&0&0\\ 0&0&1&1&1&0&0\\ 0&0&0&1&1&1&0\\ 0&0&0&1&1&0&0\\ 0&0&1&1&0&0&0\\ 0&1&1&0&0&0&0\\ 1&1&0&0&0&0&0 \end{bmatrix} * \begin{bmatrix} 1&0&1\\ 0&1&0\\ 1&0&1 \end{bmatrix} = \begin{bmatrix} 1&4&3&4&1\\ 1&2&4&3&3\\ 1&2&3&4&1\\ 1&3&3&1&1\\ 3&3&1&1&0 \end{bmatrix}$$

Figure 3: An example illustrates the convolution operation.

2.2 Pooling layers

The pooling (sub-sampling) layer is used to reduce the resolution of the features. It keeps the features from noise and distortion. There are two methods can be calculated to do pooling: max pooling and average pooling. In both cases, the input is divided into non-overlapping two-dimensional spaces. For example, as showed in Figure 1, the second layer of feature map 14x14 as input that uses the max polling of size region 5x5 is the pooling layer. For max pooling, the maximum value of the twenty five values that correspond to specified area of input is selected. While for average pooling method is selected, the average of the twenty five values in the region is calculated. Figure 4 show how the pooling process can be calculated using both methods. Assume the input is of size 4x4 and 2x2 for sub-sampling region, a 4x4 image is divided into four non-overlapping matrices of size 2x2. In the case of using max pooling method, the maximum value of the four values in the 2x2 matrix is obtained. In case of average pooling method when selected, the average of the four values is calculated. Please note according to this example that for the output with index (2, 2), the calculated result of averaging is a fraction that has been rounded to nearest integer.



21	8	8	12
12	19	9	7
8	10	4	3
18	12	9	10

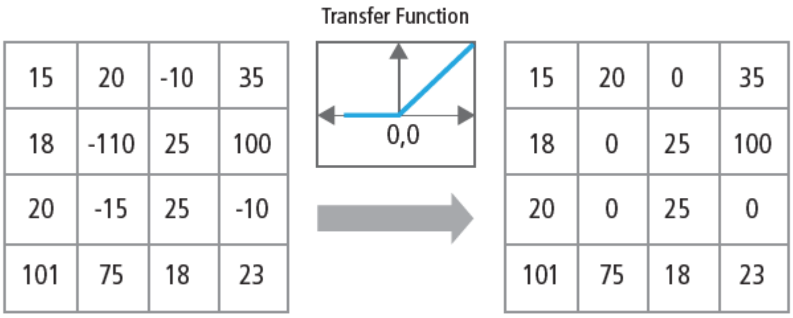
15	9
12	7

21	12
18	10

Figure 4: graphical representation of how to use max pooling and average pooling methods

2.3 Activation layers (Rectified linear units (ReLUs))

CNNs may use a variety of specific non-linear “trigger” functions (between the two layers of convolution - Maxpool and on each hidden layer), such as rectified linear units (ReLUs) as shown in figure 1 or continuous trigger (non-linear) functions (e.g., hyperbolic tangent, and sigmoid) to efficiently implement this non-linear triggering. CNN prefers RelUs because the network trains using it many times faster compared to the other non-linear functions. The function of the RELU can be implemented by y = max(x,0), so the input and output sizes of this layer are the same. It increases the nonlinear properties of the overall network without affecting the accessible fields of the convolution layer. Figure 5 show as an example how the ReLU with its transfer function can be operated.



15	20	-10	35
18	-110	25	100
20	-15	25	-10
101	75	18	23

15	20	0	35
18	0	25	100
20	0	25	0
101	75	18	23

Figure 5: graphical representation of ReLU functionality

2.4 Fully Connected Layers

As described above according to showing of figure 1, a typical CNN consists of several convolutional layers where each output of convolution layer is processed by a pooling layer. The last building block of CNN can be represented by the fully connected layer, which is basically a traditional MLP. Once the features are extracted, they are mapped by a subset of FC layers to calculate the final outputs of the network. After feature extraction the data can be classified into various classes using this layer (FC). The output feature maps of the final convolution or pooling layer is typically transformed into a one dimensional (1D) array of numbers, that is connected to one or more fully connected layers, also defined as dense layers. Where every input is connected to every output to be calculated based on an updatable weight multiplied by each input with accumulated or summation for all inputs connected to each one of output node. The final fully connected layer typically has the same number of output nodes as the number of classes. Each fully connected layer is followed by a nonlinear function, such as ReLU, as shown in figure 1. i.e. Fully connected layers mathematically sum a weighting of the previous features layer, indicating the precise mix of feature and weighting elements to determine a specific target output result [20]. To illustrate this, figure 6 explains as an example of how the fully connected layer L is calculated. Previous Layer L-1 $\{Y_0,Y_1\}$has two features, each of which is 2x2, i.e., has four elements that are multiplied by waiting element 2x2 $\{w_{00},w_{10},w_{01},w_{11}\}$ and summed (Accumulation) to produce an output of current Layer L $\{Y_0,Y_1\}$has two features, each having a single element.

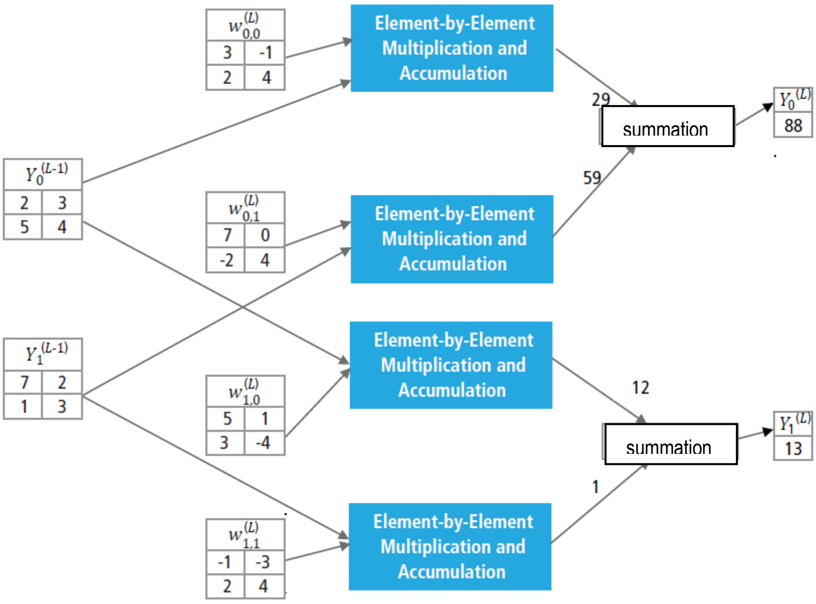

$w_{0,0}^{(L)}$
3 -1
2 4
Element-by-Element Multiplication and Accumulation
29
summation
$Y_0^{(L)}$
88
$Y_0^{(L-1)}$
2 3
5 4
59
$w_{0,1}^{(L)}$
7 0
-2 4
Element-by-Element Multiplication and Accumulation
Element-by-Element Multiplication and Accumulation
$Y_1^{(L-1)}$
7 2
1 3
$w_{1,0}^{(L)}$
5 1
3 -4
12
summation
$Y_1^{(L)}$
13
1
$w_{1,1}^{(L)}$
-1 -3
2 4
Element-by-Element Multiplication and Accumulation

Figure 6: show how a fully connected layer is being processed

2.5 Dropout

Dropout is a simple method to prevent neural networks from overfitting. i.e. Large neural nets trained on relatively small datasets (predefined data) can overfit. This has the effect of the model learning, which results in poor performance when the model is tested with new data. The Dropout layer is a mask that neglects the operation of some neurons towards the next layer and leaves it as unmodified or not connected to the others. A Dropout layer can be applied to the input vector, in which case it nullifies some of its features; also it can be applied to a hidden layer, in which case it nullifies some hidden neurons as shown in figure 7.

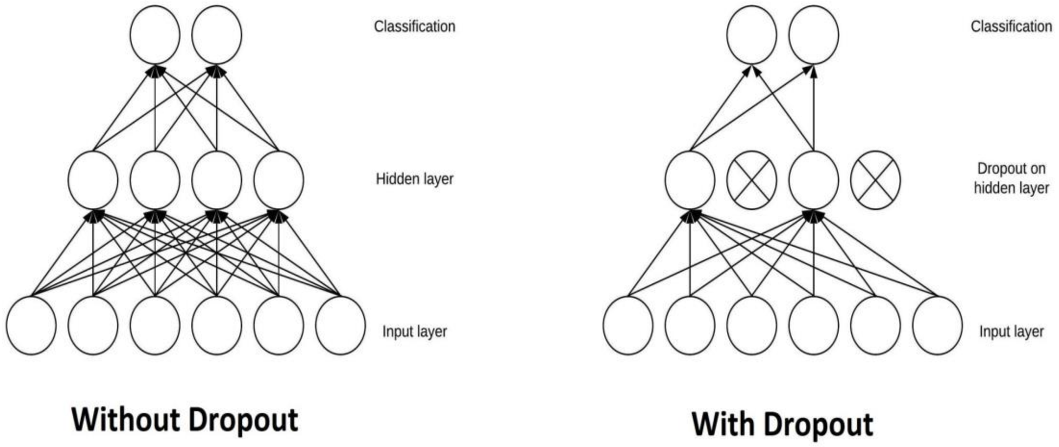


Figure 7: fully connected layer without /with dropout

2.6 SoftMax Unit

A SoftMax unit is usually used as the output of the fully connected layer as shown in figure 8. A SoftMax unit uses SoftMax function $f_j\,(z)$ to represent the probability distribution of k classes where j from 1 to k. The SoftMax function $f_j\,(z)$was used to compute the probabilities of each class. The sum of the probabilities of all classes must be equal to one. The SoftMax can be computed as (1).

$$f_j\,(z) = \frac{e^{z_j}}{\sum_k e^{z_k}} \tag{1}$$

where $f_i(z)$ is the output of the i^th of k-way SoftMax unit, SoftMax ($f_j\ (z)$) is the probability that the input instance is in class j, and z_j is the j^th element of the vector $z_i\ , i=1\ to\ k$.
give an example for illustration as in figure 8, assume two classes with output $\{z_1,z_2\}$where the SoftMax function is calculated $\{f(z_1),f(z_2)\}$and the maximum value of probability can be selected to represent the decision of predicted class.

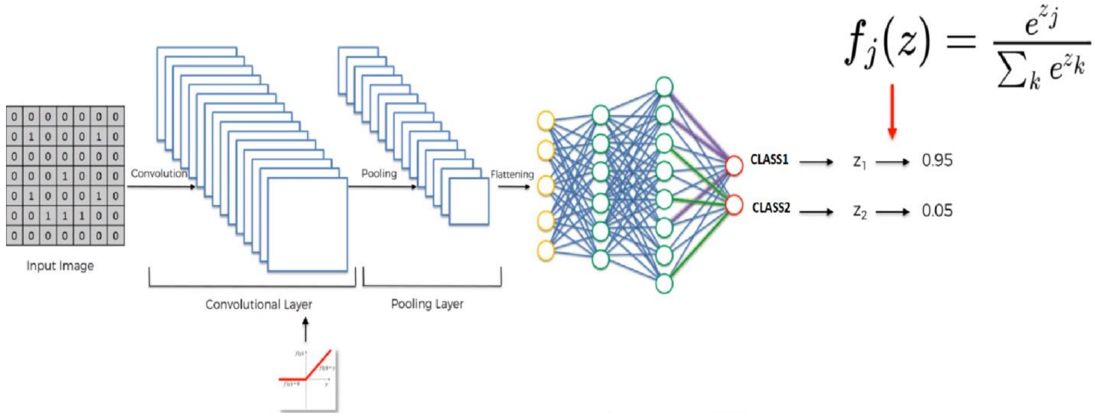


Figure 8: calculation of SoftMax function at the end of fully connected layer

3. **CNN MODELS**

CNN model is an advanced technique that appeared to deal with large databases and composite systems. This model automatically is able to extract the features of an image. Recently, fast improvements have been established in using this technique. These improvements include changing to parameter in the architecture of CNN layers for optimization and for increasing the performance which are obtained from the rearrangement of the processing units and depth of the network layers .In this section the various popular CNN models are discussed with producing a summary for all presented architectures as described below.

LeNet: It is the first CNN model that produced the state-of-art performance [21]. This model classified the digit regardless of scale, position, and distortions. It consists of 7 layers' architecture which includes 5 convolution and two pooling layers, followed by 2 fully connected layers.

AlexNet: This model obtained the cutting-edge result for image classification [22]. It was the first CNN, trained on two graphical processing units (GPUs) to decrease the hardware processing time. It was reach up to 8 layers to make it applicable on various categories of images. Moreover, filters size at initial layer were changed to 11x11 and 5x5 compared to earlier proposed network and ReLU activation function was used to overcome the issue of vanishing gradient.

ZfNet: This model increases the performance of CNN by the absence of knowledge in case of composite images [23]. In this model the feature visualization is experimentally tested against model AlexNet, which viewed that very decreased number of neurons were active in the first and second layer. So, this leads to reduce in size of filters to maximize the learning rate and to maintain the maximum feature collection in the initial two layers.

VGG-19: This model is the easiest way to increase the performance with depictive capacity of the network. This model contain 19-layer deep network (as represented with example in figure 11) compared to previous models of ZfNet and AlexNet. Moreover max pooling is used for network tuning after the convolution layer [24 -25].

GoogleNet: This model is also known as Inception-V1 was designed to achieve more accuracy with less computational cost [26]. It integrated the concept of multi-scale convolutional transformation with unify transform, and split. The convolution layer of this model structure was changed [27]. The filters of the layer block with size of 3x3, 5x5 and 1x1 to get the specified information at different level.

ResNet-34, 50, 101,152: The first ResNet architecture was represented by the Resnet-34, it have fewer filters and lower complexity. This model achieves a higher performance compared to VGG-19. Resnet-34 consisted of 34 weighted layers illustrated by given example in figure 11. While the Resnet-50 architecture is based on the above model Resnet34 [28], the major difference between them, each of the 2-layer blocks in Resnet-34 was replaced with a 3-layer bottleneck block, to be the ResNet-50 architecture. This model has much higher accuracy than the 34-layer ResNet model. Another models invented such as ResNet-101 and ResNet-152 Architecture contain Large Residual Networks. These models are constructed by using more 3-layer blocks. And even at increased network depth, the 152-layer ResNet has much lower complexity than VGG-19 nets.

R-CNN, This model is a Convolutional Neural Network (CNN) combined with a region-proposal algorithm that hypothesizes object locations. It initially extracts a fixed number of regions (2000), by means of a selective search. It then merges similar regions together, using a greedy algorithm, to obtain the candidate regions on which the object detection will be applied. Afterwards, the same authors proposed an enhanced algorithm called Fast R-CNN as shown in figure 10, by using a shared convolutional feature map that the CNN generates directly from the input image, and from which the regions of interest (RoI) are extracted [38].

YOLOvx: It is an acronym for You Only Look Once, does not extract region proposals, but processes the complete input image only once using a Fully Convolutional Neural Network that predicts the bounding boxes and their corresponding class probabilities, based on the global context of the image. The first version was published in 2016. Later on in 2017, a second version, YOLOv2, was proposed, which introduced batch normalization, a retuning phase for the classifier network, and dimension clusters as anchor boxes for predicting bounding boxes. Finally, in 2018, YOLOv3 as shown in figure 9, improved the detection further by adopting several new features [38].

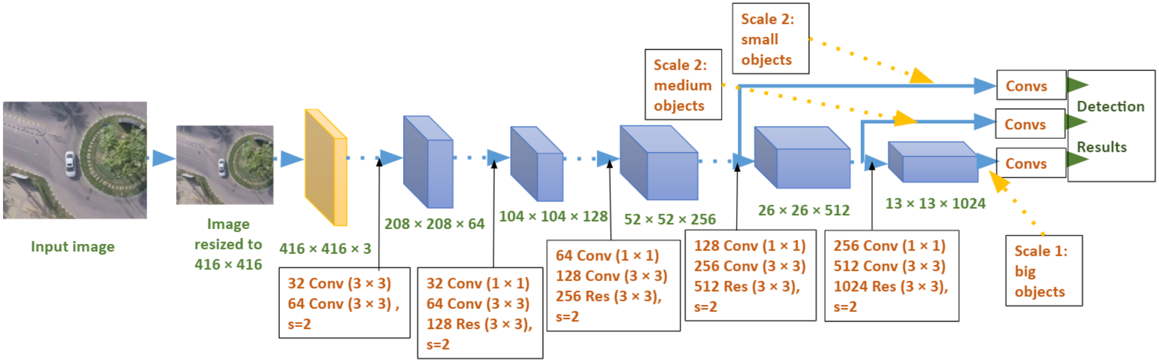

Scale 2: small objects
Scale 2: medium objects
Convs
Convs
Convs
Detection Results
Input image
Image resized to 416 × 416
416 × 416 × 3
208 × 208 × 64
104 × 104 × 128
52 × 52 × 256
26 × 26 × 512
13 × 13 × 1024
32 Conv (3 × 3) 64 Conv (3 × 3) , s=2
32 Conv (1 × 1) 64 Conv (3 × 3) 128 Res (3 × 3), s=2
64 Conv (1 × 1) 128 Conv (3 × 3) 256 Res (3 × 3), s=2
128 Conv (1 × 1) 256 Conv (3 × 3) 512 Res (3 × 3), s=2
256 Conv (1 × 1) 512 Conv (3 × 3) 1024 Res (3 × 3), s=2
Scale 1: big objects

Figure 9: YOLOv3 architecture

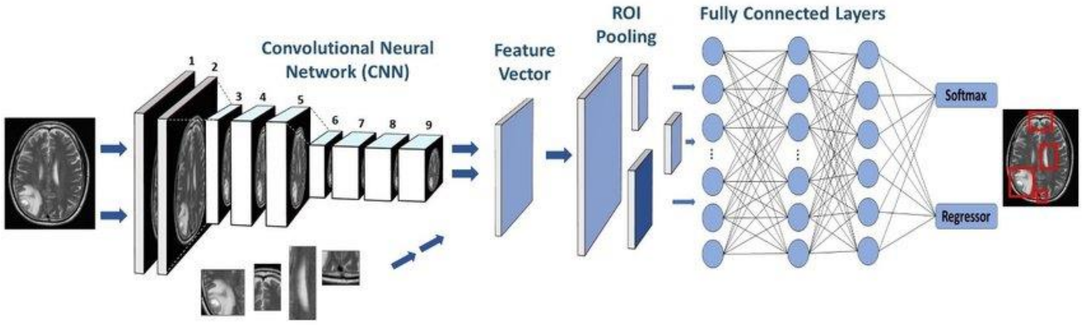


Figure 10: the building block of fast-R-CNN

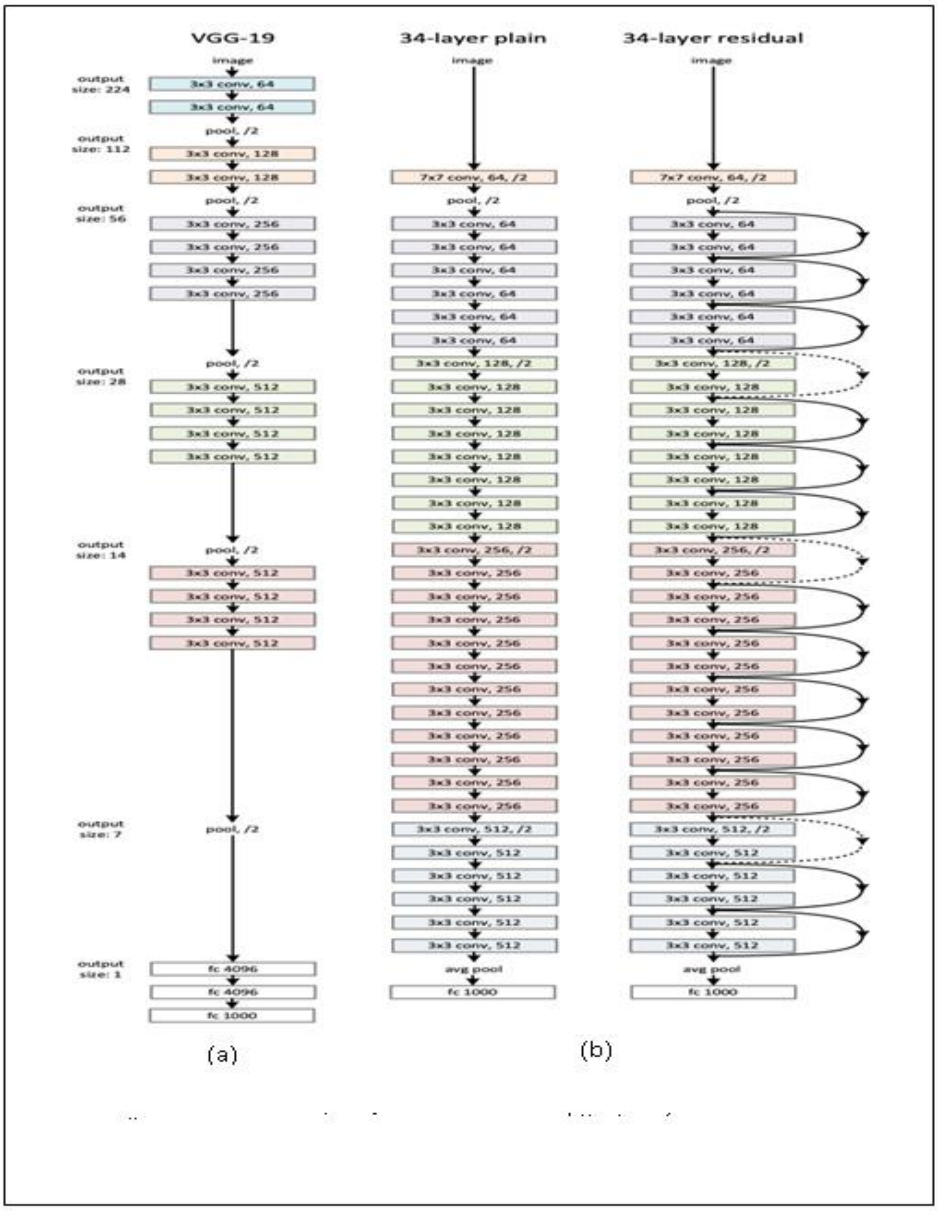


Figure 11: Some examples of network model architecture (a) VGG-19 model (b) ResNet-34 model.

For summarization of this section, the following table 1 represents the different CNN models with their feature and building blocks.

Table 1: A Summary of CNN Models

Model	**Features**	**CNN building blocks**
LeNet-5	One of the earliest CNN, Spatial features are extracted, Tanh is used as the activation function, and MLP is used as a last layer for classification.	7 layers (5 convolutional, 2 pooling layers), 2 fully connected layers.
AlexNet	First Deep CNN network, Improves the generalization for images, ReLU is the activation function, Overfitting with more number of layers, trained on almost 15 million images.	8 Layers (5 convolutional, 3 max-pooling layers), 3 fully connected layers. It uses 5x5 kernels.
ZFNet	An extension of AlexNet, ReLU is the activation function, trained on 1.3 million images.	Same as AlexNet except it uses 7x7 kernels.
GoogleNet	Reduced the computation complexity using Inception module.	22 layers with Inception Modules.
VGGNet-16,19	ReLU is the activation function, Model is simple but slow for training.	16 or 19 convolutional Layers ended with 2 or 3 fully connected layers followed by a SoftMax layer for output.
ResNet-50 ,101,152	Resolved the problem of vanishing gradient, higher performance for classification, detection, and localization tasks.	At least 50 or 101 or 152 convolutional layers ended with 1or 2 fully connected layer.
R-CNN	Models in the R-CNN family are all region-based. The detection happens in two stages: (1) First, the model proposes a set of regions of interests The proposed regions are sparse as the potential bounding box candidates can be infinite. (2) Then a classifier only processes the region candidates.	The R-CNN takes the feature map for each proposal, flattens it and uses two fully-connected layers of size 4096 with ReLU activation.
YOLOvx	The YOLO model is the very first attempt at building a fast real-time object detector. Because YOLO does not undergo the region proposal step and only predicts over a limited number of bounding boxes, it is able to do inference super-fast.	The base model is similar to GoogLeNet with inception module replaced by 1x1 and 3x3 conv layers. The final prediction of shape is produced by two fully connected layers over the whole conv feature map.

4. APPLICATIONS OF CONVOLUTIONAL NEURAL NETWORKS IN AI

This study focused on the applications in AI using CNNs to enhance the lives of people. Simple applications of CNNs which can be seen in everyday life are clear choices, such as face recognition, speech recognition, image detection, pattern classification, etc. Some of the key applications of CNN are listed below:

4.1 Face Recognition

Face recognition can be divided by a convolutional neural network into the following major components: detecting the feature of face object in the capture of picture, Focusing on each face in spite of external factors, such as light, angle, pose, etc., Identifying unique features to be compared with already existing data in the database to match a face with identity. In this type of application, a CNN can be used for face recognition. The CNN network is composed of two convolution layers, a fully connected layer, and a classification layer. Each convolution layer is followed by an activation layer and a MaxPooling layer. In additional to, two regularization techniques are added after each convolution layer: dropout and batch normalization. After the fully connected layer, the dropout technique is applied to enhance the performance of the CNN network and to reduce the complexity of calculations. Using the CNN in this type of application achieves a training accuracy of 99.78%, a validation accuracy of 98.7%, based on an assumption speed of 231 frames per second [29].

4.2 Understanding Climate

CNNs can be used to know how the climate change can be detected, and understanding the reasons of why the changes occur. This is not easy for humans to detect the changes, and there is a need for more manpower to carry out deeper experiments in this field. CNNs can provide powerful tools for identifying, classifying, and predicting patterns in climate. The deep CNN trained with 3000 samples or more per cluster has an accuracy of 94%. The architecture of CNN in this type of application has 4 convolutional layers that have 8, 16, 32 and 64 filters, respectively. Each filter has a kernel size of 5 × 5, and filters size of the max-pooling layer have a kernel size of 2 × 2. Each convolution step is followed by the ReLU layer. The extracted features maps into a single (1D) array which is connected to a fully connected layer with 1024 neuron [30].

4.3 Advertising

In This type of application, a CNN model called advert-detection mode (ADNet) is proposed, that detects the presence of advertisements automatically in video frames. This is motivated from the VGG-19 architecture. The VGG-19 model uses small convolution filters (3x3), with depth up to 19 weight layers. The VGG-19 has 6 different configurations, depending on the number of weight layers. The ADNet model has a classification accuracy of 0:94 [31].

4.4 Optical Character Recognition

Accurate optical character recognition (OCR) software can be developing. It is based on digitizing the existing documents into text at a high accuracy rate and applies them for further analysis. The OCR takes an optical image of character as input and produces the corresponding text character as output. It has a wide range of applications including robotics, traffic surveillance, etc. The OCR can be implemented by using CNN architecture. The traditional CNN classifiers are capable of learning the images contents and classify them; the classification is performed by using SoftMax layer. In this type of application, OCR is presented by combining CNN architecture and Error Correcting Output Code (ECOC) classifier. The CNN is used for feature extraction and the ECOC is used for classification. There are different CNN architectures for classification. One of the most popular deep CNN networks include AlexNet model. This model accepts input of size 227X227X3. It contains 5 convolution layers which are followed by ReLU layer and Max Pooling layer. The first two convolution layers are also followed by cross channel normalization layer. It also contains three fully connected layers followed by ReLU Layer and the first two fully connected layers are also followed by dropout layers. The output of the last fully connected layer is given as an input to SoftMax which produces the probability distribution of approximately 1000 classes. The AlexNet model was designed for recognizing objects in Image-Net which can takes an RGB image of size 128 × 128 × 3 as input. A training accuracy rate of 98.97% and testing accuracy of 97.06% have been achieved [32].

4.5 Object Recognition

Object recognition software can be developed that runs powerful deep learning algorithms for image recognition. This software can help you to classify different objects based on their color, size, type, etc. YOLOv3 model and faster region convolutional neural network (FAST-RCNN) are deep learning object detection approach of CNN. The main goal is to improve the classification accuracy. YOLOv3 uses a new fully convolution feature extraction network Darknet-19 model, which contains a total of 19 convolutional layers and 5 maximum pooling layers. By adding a batch normalization layer to the convolutional layer and removing dropout. The accuracy of this network reaches to value of 96:07% [33].

4.6 Barcode/QR Code Recognition

Software for scanning each barcode or QR code of any product can be built which will take images of the product, scan the barcode/QR code and display the corresponding information by the system. Fast-RCNN model which is an object detection method based on CNN is used, this contains some basic convolution, RelU and pooling layers to extract feature maps of the input image. Convolution layers generally accept convolution kernel with the size of 3*3, and each pooling layer reduces the picture to half of the original size. Training the Fast-RCNN model made the accuracy to be 90.77% [34].

4.7 Emotion Analysis Software

This software can be build which will help in recognizing emotions based on extracting features of people's faces and use it for analyzing and making decisions. A novel technique called face emotion recognition (FERC) is introduced using CNNs. The FERC is based on two-part convolutional neural network: The first part removes the background from the picture, and the second part concentrates on the facial feature vector extraction. In this model, expressional vector (EV) is used to recognize the five different types of regular face expression. The main advantage of the model FERC is that it adopt with different orientations (less than 30°). FERC model is a unique developed method with two 4-layer and four filters. This networks with an accuracy reach to 96% [35].

4.8 Defect Recognition Software

This software has the capability to detect any defects that refer to the specified product by analyzing the images of the product on the conveyor belt. This helps in reducing misuse among your customers by consistently providing high-quality products. CNN can be used with general detecting method to overcome the common shortcoming. CNN is used to complete image patch classification. And features are automatically exacted in this part. Then, an adding mechanism can be build to exclude the interference of the background image, realizing classification and location. In CNN structure, it takes image blocks as input with size of 64 x 64. Firstly, network automatically extracted features based on some processes of the block which contains several convolution and pooling layers. Then, inception layer raised from GoogleNet is used to increase network performance. Inception layer is consisted of four branches, three of them are convolution kernels with size 1 x 1, 3 x 3, 5 x 5 and the rest branch is a 3 x 3 pooling kernel. Each branch extracts features from the upper layer. Then features are concatenated together. This CNN model reaches to an accuracy of 98.6% [36].

4.9 Satellite Image Analysis

Software developers can develop satellite image recognition and analysis software at highly reasonably priced. This work introduces a competitive methodology for crop classification from multispectral satellite imagery. This mainly using an enhanced 2D convolutional neural network (2D-CNN) designed at smaller-scale architecture, in additional to a novel post-processing step. This contains four steps: image stacking, patch extraction, classification model design (based on a 2D-CNN architecture), and post-processing. Firstly, the 2D-CNN is trained independently for each one of the sequences by extracting patches of 32 x 32 x n pixels in size, where n is the number of bands in the sequence. Secondly, the training patches are passed through three layers of convolution, to provide the feature maps. Thirdly, a pooling layer reduces the feature map size. Finally, the classification task itself is established by a fully-connected layer and the output layer. This methodology achieves a competitive accuracy of 81.20%, and an average accuracy of 88.72% when classifying 10 different crops [37].

4.10 Other Interesting Fields

CNNs are pointed to be the future with their introduction into self driving cars, robots that can simulate human behavior, predicting earthquakes, natural disasters, and self-diagnoses of medical problems. One problem that researchers are working on with CNNs is brain cancer detection. The earlier detection of brain cancer can establish a big step in saving more lives affected by this illness.

5. CONCLUSION

Convolutional Neural Network (CNN) is one technique of deep learning can be implemented using a software programming language that helps computers to understand the features or contents exist in images, speech and videos with higher accuracy. These are required in many applications of artificial intelligence (AI). The implementation of these applications based on understanding the architecture of CNNs models which have four fundamental building blocks have been described. This description of CNN model architecture includes defining the function of each block and how CNN work in different applications. Most popular models of CNN have been reviewed. Moreover, a review of the work carried for applying the described CNN models used in the interesting applications which are daily exist in our life. Finally the aim of this paper is to provide overview of CNN building block architecture and illustrate how CNN models are useful for different AI applications.

REFERENCES

[1] R. Ayachi, M. Afif, Y. Said, M. Atri, “Traffic signs detection for real world application of an advanced driving assisting system using deep learning”, Neural Processing Letters, Vol. 51, 2020, pp. 837-851.

[2] R. Ayachi, Y. E. Said, M. Atri, “To perform road signs recognition for autonomous vehicles using cascaded deep learning pipeline”, Artificial Intelligence Advances, Vol. 1, No. 1, 2019, pp. 1-58.

[3] M. Afif, R. Ayachi, Y. Said, E. Pissaloux, M. Atri, “Indoor image recognition and classification via deep convolutional neural network”, 8th International Conference on Sciences of Electronics, Technologies of Information and Telecommunications, Maghreb, Tunisia, December, 2018,pp.18-20.

[4] M. Afif, R. Ayachi, Y. Said, E. Pissaloux, M. Atri, “An evaluation of Retinanet on indoor object detection for blind and visually impaired persons assistance navigation”, Neural Processing Letters, Vol. 51, 2020, pp. 1-15.

[5] M. Afif, R. Ayachi, Y. Said, E. Pissaloux, M. Atri, “Indoor object c1assification for autonomous navigation assistance based on deep CNN model”, IEEE International Symposium on Measurements & Networking, Catania, Italy, July , 2019, pp. 8-10.

[6] D. Virmani, P. Girdhar, P. Jain, P. Bamdev, “FDREnet,” Face detection and recognition pipeline”, Engineering, Technology & Applied Science Research, Vol. 9, No. 2, 2019 pp. 3933-3938.

[7] U. Khan, K. Khan, F. Hasssan, A. Siddiqui, M. Afaq, “Towards achieving machine comprehension using deep learning on non-GPU machines”, Engineering, Technology & Applied Science Research, Vol. 9, No. 4, 2019 pp. 4423-4427.

[8] Dong Yu and Li Deng. “Automatic speech recognition”. Springer, 2016.

[9] Monita Chatterjee, Danielle J Zion, Mickael L Deroche, Brooke A Burianek, Charles J Limb, Alison P Goren, Aditya M Kulkarni, and Julie A Christensen, “Voice emotion recognition by cochlear-implanted children and their normally-hearing peers”, Hearing research, vol. 322, 2015, pp.151–162.

[10] Nancy Eisenberg, Tracy L Spinrad, and Natalie D Eggum. “Emotion-related self-regulation and its relation to children’s maladjustment”. Annual review of clinical psychology, vol. 6, 2010, pp. 495–525..

[11] Ciresan, Dan, Ueli Meier, and Jürgen Schmidhuber. “Multi-column deep neural networks for image classification.” 2012 IEEE Conference on Computer Vision and Pattern Recognition (New York, NY: Institute of Electrical and Electronics Engineers (IEEE)). 2012..

[12] Ciresan, Dan, Ueli Meier, Jonathan Masci, Luca M. Gambardella, and Jurgen Schmidhuber. “Flexible, High Performance Convolutional Neural Networks for Image Classification.” Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence-Volume Two: 1237–1242. 2011. Retrieved 17 November 2013.

[13] Lawrence, Steve, C. Lee Giles, Ah Chung Tsoi, and Andrew D. Back. “Face Recognition: A Convolutional Neural Network Approach.” IEEE Transactions on Neural Networks, Volume 8; Issue 1. 1997.

[14] Russakovsky, O. et al. “ImageNet Large Scale Visual Recognition Challenge.” International Journal of Computer Vision, 2014. .

[15] Alex Krizhevsky, Ilya Sutskever, and Geo_rey E Hinton.” ImageNet classification with deep convolutional neural networks”. In Advances in neural information processing systems, 2012, pp. 1097-1105.

[16] Steve Lawrence, C Lee Giles, Ah Chung Tsoi, and Andrew D Back. “Face recognition: A convolutional neural-network approach”. IEEE transactions on neural networks, vol. 8(1), 1997, pp. 98-113,

[17] Yann LeCun, Bernhard E Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne E Hubbard, and Lawrence D Jackel. “Handwritten digit recognition with a back-propagation network”. In Advances in neural information processing systems, 1990, pp. 396-404.

[18] Wayne Xiong, Jasha Droppo, Xuedong Huang, Frank Seide, Mike Seltzer, Andreas Stolcke, Dong Yu, and Geo_rey Zweig. “The Microsoft 2016 conversational speech recognition system”. In Acoustics, Speech and Signal Processing (ICASSP), IEEE International Conference, 2017, pp. 5255-5259.

[19] Xiang Zhang, Junbo Zhao, and Yann LeCun. “Character-level convolutional networks for text classification”. In Advances in neural information processing systems, 2015, pp. 649-657.

[20] Yann LeCun, Yoshua Bengio, et al. Convolutional networks for images, speech, and time series. The handbook of brain theory and neural networks, 3361(10): 1995.

[21] Y. LeCun et al., “Learning algorithms for classification: A comparison on handwritten digit recognition,” Neural networks Stat. Mech. Perspect., vol. 261, 1995.

[22] Krizhevsky, Alex & Sutskever, Ilya & Hinton, Geoffrey, “ImageNet Classification with Deep Convolutional Neural Networks”. Neural Information Processing Systems: 2012.

[23] Zeiler, Matthew & Fergus, Rob, “Visualizing and Understanding Convolutional Neural Networks”, ECCV 2014.

[24] Simonyan, Karen & Zisserman, Andrew, “Very Deep Convolutional Networks for Large-Scale Image Recognition”, arXiv, 2014, pp.1409.1556,

[25] M. Ranzato, F. J. Huang, Y. Boureau and Y. LeCun, "Unsupervised Learning of Invariant Feature Hierarchies with Applications to Object Recognition," 2007 IEEE Conference on Computer Vision and Pattern Recognition, Minneapolis, MN, 2007 pp. 1-8.

[26] C. Szegedy et al., "Going deeper with convolutions," 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, 2015, pp. 1-9.

[27] Y. Pang, M. Sun, X. Jiang and X. Li, "Convolution in Convolution for Network in Network," in IEEE Transactions on Neural Networks and Learning Systems, vol. 29, no. 5, May 2018, pp. 1587-1597.

[28] K. He, X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, 2016, pp. 770-778.

[29] Yahia Said, Mohammad Barr, Hossam Eddine Ahmed, “ Design of a Face Recognition System based on Convolutional Neural Network (CNN)” , Engineering, Technology & Applied Science Research Vol. 10, No. 3, 2020, pp. 5608-5612,

[30] Ashesh Chattopadhyay, Pedram Hassanzadeh, Saba Pasha, “Predicting clustered weather patterns: A test case for applications of convolutional neural networks to spatio-temporal climate data”, Scientific Reports *10:1317*|, *2020,*

[31] Murhaf Hossari, Soumyabrata Dev, Matthew Nicholson, Killian McCabe1, Atul Nautiyal, Clare Conran, Jian Tang, Wei Xu, and Fran_cois Piti_e;” ADNet: A Deep Network for Detecting Adverts”, CEUR-WS.ORG/VOL-2259\aics_6.pdf.

[32] Mayur Bhargab Bora, Dinthisrang Daimary, Khwairakpam Amitab_, Debdatta Kandar “ Handwritten Character Recognition from Images using CNN-ECOC”, International Conference on Computational Intelligence and Data Science, Published by Elsevier B.V., Procedia Computer Science vol.167, 2020,pp. 2403–2409.

[33] Rafael Magalhães, Maia Peixoto,”Object Recognition Using Convolutional Neural Networks” In book: Artificial Neural Networks, November 2019..

[34] Jinbo Peng, Song Yuan, and Xin Yuan, “QR Code Detection with Faster-RCNN Based on FPN” © Springer Nature Switzerland AG 2020 X. Sun et al. (Eds.): ICAIS 2020, LNCS 12239, 2020, pp. 434–443.

[35] Ninad Mehendale, “Facial emotion recognition using convolutional neural networks (FERC)” © Springer Nature Switzerland AG, 2020.

[36] Prahar M. Bhatt†, Rishi K. Malhan†, Pradeep Rajendran†, Brual C. Shah†, Shantanu Thakar†, Yeo Jung Yoon†, and Satyandra K. Gupta “Image-based Surface Defect Detection Using Deep Learning: A Review”,Journal of Computing and Information Science in Engineering , January 2021.

[37] Moreno-Revelo,M. Y.; Guachi-Guachi, L.; Gómez-Mendoza, J.B.; Revelo-Fuelagán, J.; Peluffo-Ordóñez, D.H. “Enhanced Convolutional-Neural- Network Architecture for Crop Classification”. Appl. Sci., 11, 4292, 2021.

[38] Ammar, A.; Koubaa, A.; Ahmed, M.; Saad, A. Aerial Images Processing for Car Detection using Convolutional Neural Networks: Comparison between Faster R-CNN and YoloV3. Preprints 2019, 2019100195 (doi: 10.20944/preprints201910.0195.v1).