electronics

MDPI

Review

CNN Variants for Computer Vision: History, Architecture, Application, Challenges and Future Scope

Dulari Bhatt [1], Chirag Patel [2,*], Hardik Talsania [1], Jigar Patel [1], Rasmika Vaghela [1], Sharnil Pandya [3], Kirit Modi [4] and Hemant Ghayvat [5]

[1] Research Scholar, Parul University, Gujarat 382030, India; dulari.bos@gmail.com (D.B.); hardik.talsania@gujgov.edu.in (H.T.); jigarsharp@gmail.com (J.P.); rashmika.vaghela@gujgov.edu.in (R.V.)
[2] Computer Science & Engineering, DEPSTAR, Changa, Gujarat 388421, India
[3] Symbiosis Institute of Technology, Symbiosis International (Deemed) University, Pune 412115, India; sharnil.pandya@sitpune.edu.in
[4] Sankalchand Patel College of Engineering, Sankalchand Patel University, Visnagar 384315, India; kjmodi.fet@spu.ac.in
[5] Computer Science Department, Faculty of Technology, Linnaeus University, P G Vejdes väg, 351 95 Växjö, Sweden; hemant.ghayvat@lnu.se
* Correspondence: chiragpatel.dce@charusat.ac.in or chirag453@gmail.com

Abstract: Computer vision is becoming an increasingly trendy word in the area of image processing. With the emergence of computer vision applications, there is a significant demand to recognize objects automatically. Deep CNN (convolution neural network) has benefited the computer vision community by producing excellent results in video processing, object recognition, picture classification and segmentation, natural language processing, speech recognition, and many other fields. Furthermore, the introduction of large amounts of data and readily available hardware has opened new avenues for CNN study. Several inspirational concepts for the progress of CNN have been investigated, including alternative activation functions, regularization, parameter optimization, and architectural advances. Furthermore, achieving innovations in architecture results in a tremendous enhancement in the capacity of the deep CNN. Significant emphasis has been given to leveraging channel and spatial information, with a depth of architecture and information processing via multi-path. This survey paper focuses mainly on the primary taxonomy and newly released deep CNN architectures, and it divides numerous recent developments in CNN architectures into eight groups. Spatial exploitation, multi-path, depth, breadth, dimension, channel boosting, feature-map exploitation, and attention-based CNN are the eight categories. The main contribution of this manuscript is in comparing various architectural evolutions in CNN by its architectural change, strengths, and weaknesses. Besides, it also includes an explanation of the CNN's components, the strengths and weaknesses of various CNN variants, research gap or open challenges, CNN applications, and the future research direction.

Keywords: CNN; feature-map exploitation; attention-based CNN; deep CNN; object recognition; computer vision

Citation: Bhatt, D.; Patel, C.; Talsania, H.; Patel, J.; Vaghela, R.; Pandya, S.; Modi, K.; Ghayvat, H. CNN Variants for Computer Vision: History, Architecture, Application, Challenges and Future Scope. *Electronics* **2021**, *10*, 2470. https://doi.org/10.3390/electronics10202470

Academic Editor: Giovanni Dimauro

Received: 2 September 2021
Accepted: 25 September 2021
Published: 11 October 2021

1. Introduction

Artificial intelligence is bridging the gap between machine and human talents at a breakneck pace. Many academics and enthusiasts are working on various AI field elements to develop incredible things. One such incredible field includes the domain of computer vision. The primary goal of computer vision is to make machines see the world the same way as humans. Well-known computer vision tasks include image detection, image tagging, image recognition, image classification, image analysis, video analysis, natural language processing, and so on. Deep learning advancements in computer vision have piqued the interest of numerous academics over the years. CNN is used to construct the majority of

computer vision algorithms. A convolutional neural network is a method of deep learning that takes an input image and assigns importance (learnable biases and weights) to various objects in the image, distinguishing one from the other [1].

In comparison to other methods, CNN requires less preprocessing. Therefore, a CNN is the most effective learning algorithm for comprehending picture material [1]. Furthermore, it has demonstrated exceptional image classification, recognition, segmentation, and retrieval [2]. The accomplishment of CNN has piqued the interest of people outside of academia. Microsoft, Google, AT&T, NEC, and Facebook are among the companies engaging in the development and advancement of CNN architecture [3]. In addition, they have active research groups that are investigating novel CNN designs. At the moment, deep CNN-based models are being used by the majority of front-runners in image processing and computer vision competitions. As a result, there are several variants of the basic CNN design. This manuscript covers an introduction to CNN, the evolution of CNN over time, various features of CNN design, and the architectural analysis of each type of CNN with its benefits and drawbacks.

CNN reassembles regular neural networks, but it has an appealing characteristic made up of neurons with learnable weights and biases. Every neuron receives many inputs and then performs a dot product, which is optionally followed by nonlinearity [4]. As a result, CNN functions as a feed-forward kernel, undergoing many modifications [4]. The primary goal of convolution is to extract meaningful features from locally associated data sources. The convolutional kernels' output is then fed into the activation function, which aids in learning intellections and embeds nonlinearity in the feature space. This nonlinearity produces different activation functions for each reaction, making it easier to learn meaningful dissimilarities in images. Furthermore, a nonlinear activation function produces an output that is frequently trailed by subsampling; this supports summarizing the outputs, which makes the input insensitive to geometrical deceptions.

Najafabadi, in 2015, investigated that CNN has an automatic feature extraction capability that eliminates the requirement for a distinct feature extractor [5]. As a result, CNN can learn from a good representation of internal raw pixels without exhausting processing. Automatic feature extraction, multitasking, weight sharing, and hierarchical learning are some of CNN's appealing features [6].

CNN was formerly known as LeNet. LeNet was named after its creator, Yann LeCun. Yann LeCun created a network for handwritten digit identification in 1989, building on the work of Kunihiko Fukushima, a Japanese scientist who designed the neocognitron (essential image recognition neural network). The LeNet-5, which describes the primitive components of CNN, might be regarded as the beginning of CNN. LeNet-5 was not well-known because of hardware equipment paucity, particularly GPUs (graphics processing units). As a result, there was little research on CNN between 1990 and 2000. The success of AlexNet in 2012 opened the door for computer vision applications, and many various forms of CNNs, such as the R-CNN series, have been raised. CNN models now are quite different from LeNet, although they are all based on it.

In recent years, several exciting survey papers have been published on deep CNN. For example, (Asifullah Khan et al., 2018) examined prominent structures from 2012 to 2018 and their major components. (Alzubaidi et al., 2021) reviewed deep learning concepts, CNN architecture, challenges, and future trends. This paper was the first paper to include various DL aspects. It also includes the impact of CPU, GPU, and FPGA on various deep learning approaches. It includes one section about the introduction to CNN and its architecture. (Smarandache et al., 2019) reviewed trends in convolutional neural network architecture. This paper primarily focuses on the design of the architecture of around 10 well-known CNN models.

Similarly, there are many authors, such as Liu (2017), LeCun (2010), Guo (2016), and Srinivas (2016), who have discussed CNN's many applications and tactics [4,6–8]. As a result, this survey exemplifies the essential taxonomy discovered in the most recent and well-known CNN designs reported between 2012 and 2021. This manuscript includes

around eight major categories of CNN based on its architecture evolution. This investigation reveals the fundamental structure of CNN, as well as its roots. It also represents a wide range of CNN architectures, from their conception to their most current advancements and achievements. This survey will assist readers in developing architectural novelties in convolutional neural networks by providing a more profound theoretical knowledge of CNN design concepts.

The primary goal of this survey is to highlight the most significant components of CNN to provide readers with a comprehensive picture of CNN from a single survey paper. In addition, this survey will assist readers in learning more about CNN and CNN variants, which helps to improve the field. The contributions of this survey paper are summarized below:

This is the first review that almost provides each detail about CNN for computer vision, its history, CNN architecture designs, its merits and demerits, application, and the future work to take upon in a single paper.

- This review assists readers in making sound decisions about their research work in the area of CNN for computer vision;
- This manuscript includes existing CNN architectures and their architectural limitations, leading it to design new architecture. Furthermore, it clearly explains the merits and demerits of almost all popular CNN variants;
- It divides the CNN architecture into eight categories based on their implementation criteria, which is an exciting part of this survey paper;
- Various applications of CNN are also explained so that readers can also take on any other application area of CNN other than computer vision;
- It provides a clear listing of future research trends in the area of CNN for computer vision.

The organization of this survey paper includes the first section, which presents a methodical comprehension of CNN. Then, it explains CNN's similarities to the primate's visual brain. Section 2, the Literature Review, is divided into two subsections. The outline of the CNN components is explained in Section 2.1, and Section 2.2 describes various profound CNN architectural evolutions. It also includes CNN's eight broad categories of architectural advancements. Section 3 explores CNN applications in a variety of disciplines, Section 4 discusses current issues in CNN architecture, and Section 5 discusses the future scope of research. Finally, Section 6 concludes a survey of various CNN variants.

2. Literature Review

This section includes fundamental components of the convolutional neural network. It describes the basic architecture of CNN to further understand CNN architectural variants. It also includes various competitive recent advancements in CNN architectures.

2.1. CNN Fundamentals

CNN is a computer vision deep learning network that can recognize and classify picture features. CNN architecture was influenced by the organization and functions of the visual cortex. It is designed to resemble the connections between neurons in the human brain.

Image recognition is a task that humans have been performing from childhood. Children were taught to identify fruits and vegetables, such as apples, bananas, and watermelons. Is it possible to teach computers to perform the same thing? Is it feasible for a human to build a machine that can see and understand just like humans? The answer is yes to all of these questions. Humans must demonstrate an algorithm of millions of images before a computer can generalize the input and make predictions for images that it has never seen before, just as humans must demonstrate an algorithm of millions of images before a computer can generalize the input and make predictions for images that it has never seen before. The question is: how does the image appear to computers? Figure 1 Humans can see the rose, but computers can see the numerical data. As a result, developing a computer that can analyze and recognize images is a complex task.

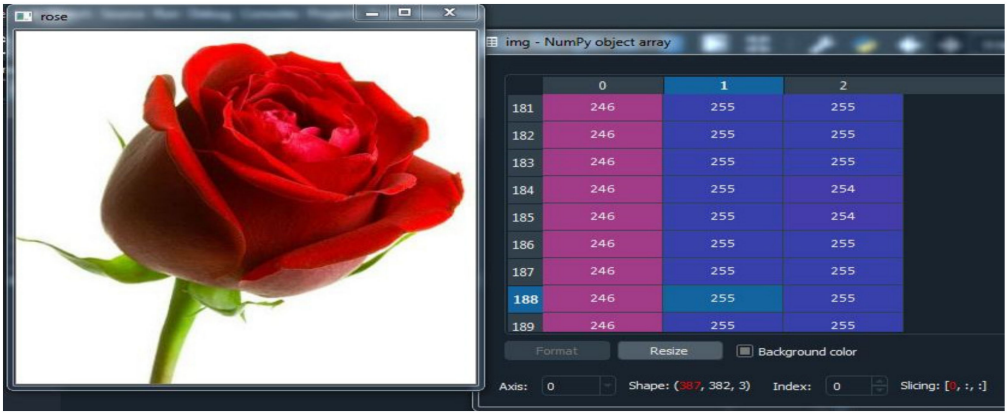


Figure 1. How computers visualize an image.

People used to add the numbers of an image of roses or, for example, target images in the database and write a program to compare the target image with the database to see if it contained a rose or a specific target image, but the main limitation was that it could not recognize any single image that was not in the database. As a result, there was a high demand for a network that automatically recognized and identified spatial properties in images.

Figure 2 shows various CNN components. To learn the advancements in CNN architecture, it is very important to understand the various CNN components and their applications.

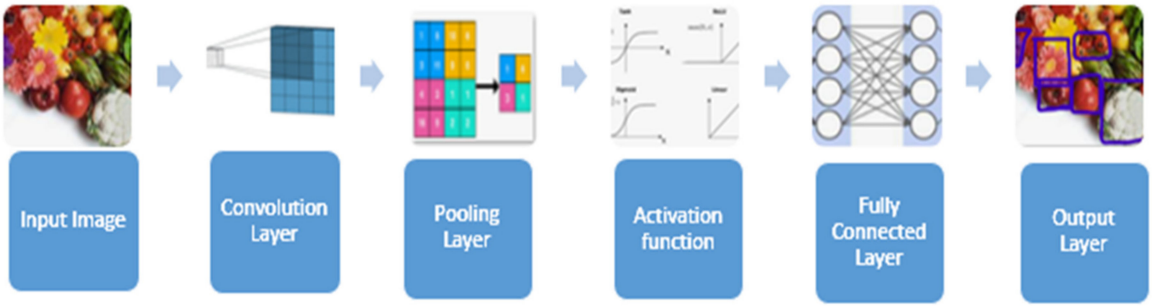


Figure 2. CNN components.

2.1.1. Input Image

Pixels are the building blocks of a computer image. They are the visual data's binary representation. A succession of pixels ranging from 0–255 are arranged in a matrix-like arrangement in the digital image. Its pixel value specifies each pixel's brightness and hue. When humans see an image, their brains process a vast amount of information in the first second. Each neuron in the human brain has its own receptive field and is connected to other neurons in order to cover the full visual field. The receptive field is a small proportion of the visual field, where each neuron in the biological vision system responds to stimuli. In the same way, each neuron in CNN analyzes data in only its receptive area. The CNN layers are programmed to identify simpler patterns first, such as lines and curves, before progressing to more complex patterns, such as faces and objects. As a result, it is plausible to claim that using a CNN may provide vision to computers.

2.1.2. Convolution Layer

The convolution layer is very important layer in the CNN architecture. It takes an image as an input and uses a 3×3 or 5×5 filter, as shown in Figure 3.

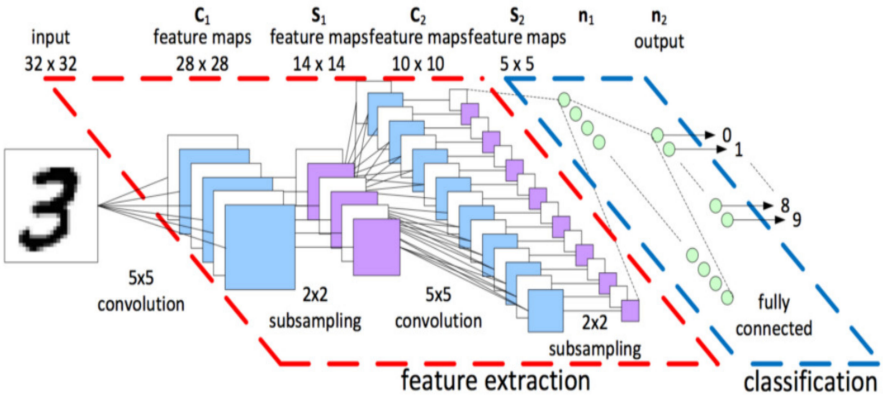


Figure 3. Convolution layer [7].

In Figure 4, the green filter slides over the input image, which is displayed in blue, one pixel at a time, starting at the top left. As it moves over the image, the filter multiplies its values with the image's overlapping values, and then adds them all together to generate a single value output for each overlap until the entire image is visited.

1	1	1	0	0
0	1	1	1	0
0	0	1	1	1
0	0	1	1	0
0	1	1	0	0

1	0	1
0	1	0
1	0	1

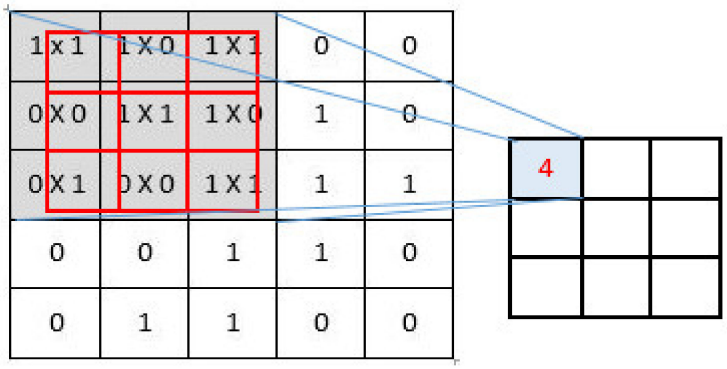
Figure 4. Input and filter image [8].

The kernel has the same depth as the input image when images have many channels, such as RGB (red, green, blue). As shown in Figure 5, matrix multiplication is conducted between the Kn and In stacks ([K1, I1], [K2,I2], [K3,I3]), and the results are then combined with the bias to produce a dense one-depth channel. Overlying receptive fields exist for each neuron in the output matrix. The first ConvLayer usually captures low-level characteristics, such as the gradient orientation, edges, color, and so forth. The design adapts to the high-level characteristics by adding layers, giving us a network with a comprehensive comprehension of the images in the dataset. Figures 5 and 6 shows the steps of convolution.

1x1	1X0	1X1	0	0
0X0	1X1	1X0	1	0
0X1	0X0	1X1	1	1
0	0	1	1	0
0	1	1	0	0

4		

Figure 5. Calculation of filter slides over input image [8].



$$1 \times 1 + 1 \times 0 + 1 \times 1 + 0 \times 0 + 1 \times 1 + 1 \times 0 + 0 \times 1 + 0 \times 0 + 1 \times 1$$

Figure 6. First step of convolution [8].

Feature Extraction

CNN is well-known for its ability to extract characteristics automatically. Figure 7 shows matrix calculation for RGB image. Padding is frequently employed in CNN in order to keep the size of the feature maps from shrinking at each layer, which is undesirable. The operation produces two types of outcomes:

1. A type in which the dimensionality of the convoluted feature is reduced in comparison to the input;
2. A type in which the dimensionality is not reduced but is either enhanced or maintained. Padding is used to satisfy this task.

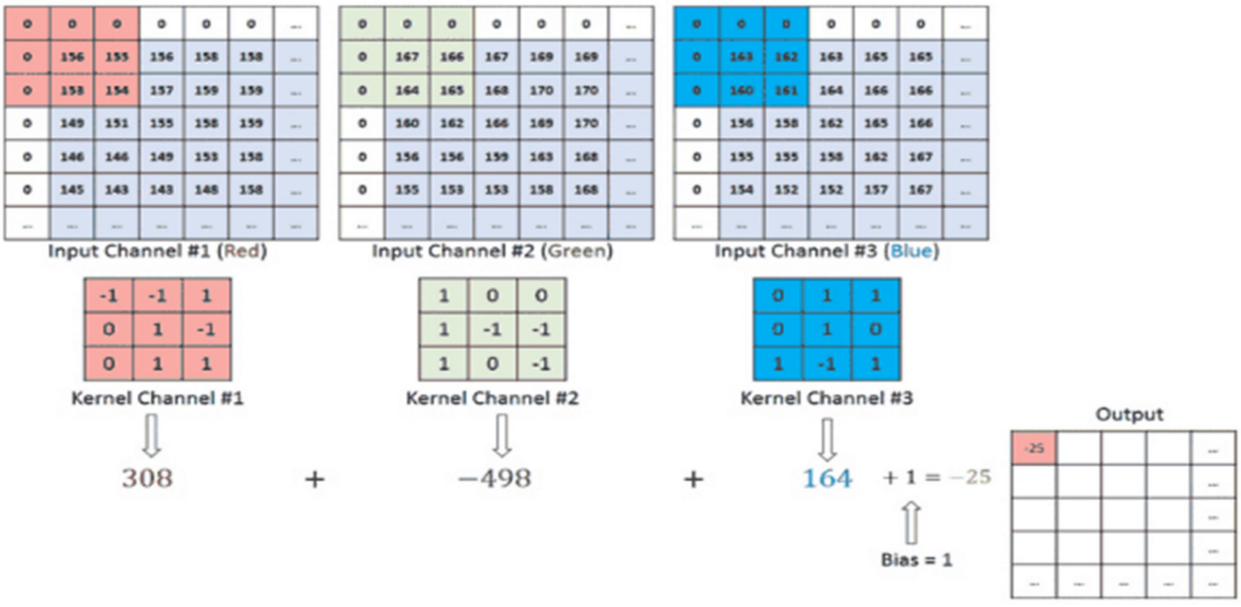


Figure 7. Matrix calculation.

For example, when the $5 \times 5 \times 1$ picture is reinforced into a $7 \times 7 \times 1$ image and then applied to the $3 \times 3 \times 1$ kernel over it, the complex matrix is observed to be of dimensions $5 \times 5 \times 1$, as shown in Figure 8. It indicates that the output image has the same dimensions as the input image (same padding). If the same procedure is conducted without padding, an image with reduced dimensions can be received in the output. As a result, a $5 \times 5 \times 1$ image will become a $3 \times 3 \times 1$ image.

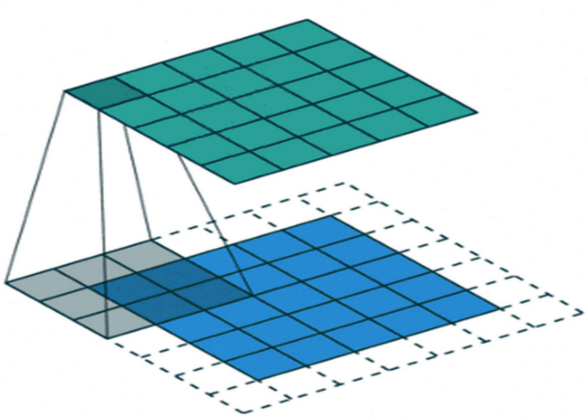


Figure 8. Padding [8].

The kernel passes through the width and height of the picture during the forwarding pass. It generates a visual representation of the receptive region in question. It generates an activation map, a two-dimensional representation of the image that shows the kernel's response at each spatial position of the image. A stride is the size of the kernel when it slides. Assume the input image is $W \times W \times D$ in size. If the number of kernels with a spatial dimension of F, stride S, and padding P is unknown, the output volume can be calculated using the following formula:

$$Wout = \frac{W - F + 2P}{S} + 1$$

This will produce an output with size Wout × Wout × Dout.

2.1.3. Pooling Layer

After obtaining the feature maps, it is necessary to add a pooling (sub-sampling) layer in CNN, next to a convolution layer. The job of the pooling layer is to shrink the convolved feature's spatial size. As a result of the dimensionality reduction, the computer power required to process the data is reduced. This also aids in the extraction of leading characteristics that are positional and rotational invariant, which preserves the model's practical training. Pooling reduces the training time while also preventing over-fitting. There are two forms of pooling: maximum pooling and average pooling.

Maximum Pooling

The tensor is the input to the pooling layer. A kernel of size n × n (2 × 2 in the aforementioned example) is moved across the matrix in the case of maximum pooling, as illustrated in Figure 9, and the maximum value is chosen and placed in the appropriate location of the output matrix.

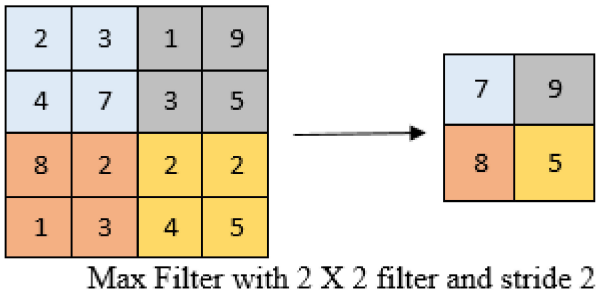


Figure 9. Max pooling [8].

Average Pooling

A kernel of size n × n is shifted across the matrix in the average pooling, and the average of all of the values is obtained for each point and placed in the corresponding position of the output matrix. This is repeated for each of the input tensor's channels. As a result, we have the output tensor. It is important to keep in mind that, while pooling reduces the image's height and breadth, the number of channels (depth) remains the same.

The pooling layer calculates a summary statistic of the surrounding outputs to replace the network output at certain points. As a result, it aids in reducing the representation's spatial dimension, which reduces the amount of computation and weights required. The pooling procedure is carried out independently on each slice of the representation. The average of the rectangle neighborhood, the L2 norm of the rectangle neighborhood, and a weighted average depending on the distance from the central pixel are all pooling functions as shown in Figure 10. The most frequent method, however, is maximum pooling, which reports the neighborhood's most significant output.

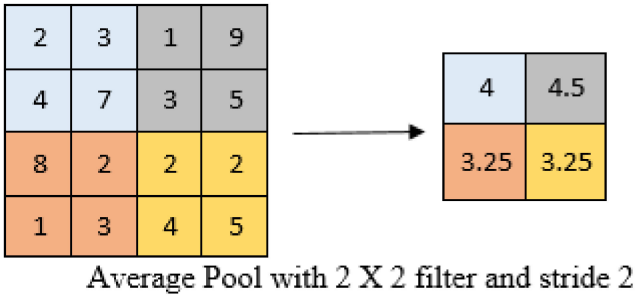


Figure 10. Average pooling [8].

2.1.4. Nonlinearity Layer (Activation Function)

The activation function plays a vital role in CNN layers. The output of the filter is provided to another mathematical function called an activation function. ReLu, which stands for rectified linear unit, is the most common activation function used in CNN feature extraction. The main motive behind using the activation function is to conclude the output of neural networks, such as yes or no. The activation function maps the output values between −1 to 1 or 0 to 1, etc. (it depends on the activation function). The activation functions can be categorized into two types:

1. Linear Activation Function This uses function F(x) = CY. It takes the input and multiplies it with constant c (weight of each neuron), and produces the output signal proportional to the input. The linear function can be better than the step function, as it only give the yes or no answer and not the multiple answers.
2. Non-linear Activation Functions In modern neural networks, non-linear activation functions are used. They enable the model to build complicated mappings between the network's inputs and outputs, which are critical for learning and modelling complex data, including images, video, audio, and non-linear or high-dimensional data sets.

2.1.5. Fully Connected Layer

A fully connected layer is nothing more than a feed-forward neural network as shown in Figure 11. Fully connected layers are found at the network's very bottom layers. A fully connected layer receives input from the final pooling or convolutional layer's output layer, which is flattened before being delivered as input. Flattening the output entails unrolling all values from the output that were obtained after the last pooling or convolutional layer into a vector (3D matrix).

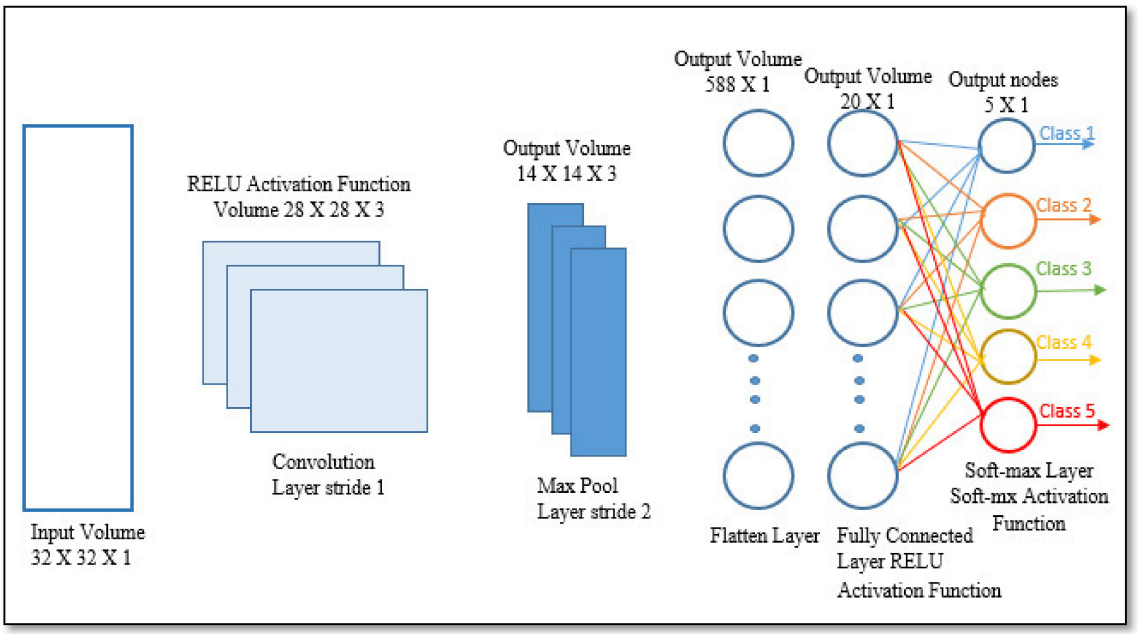


Figure 11. Fully connected Layer [8].

Adding an FC layer is a simple technique to learn nonlinear combinations of high-level features represented by the convolutional layer's output. In that space, the FC layer is learning a possibly nonlinear function.

2.2. Architectural Evolution of Deep CNNs

Figure 12. Describes the various architectural categories of CNN variants. This section explains those all categories in detail.

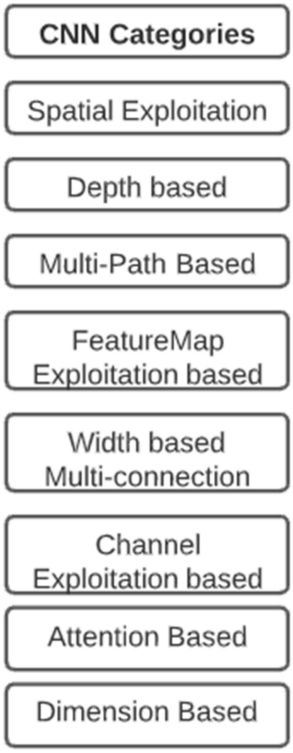


Figure 12. CNN variants categories.

2.2.1. Spatial Exploitation-Based CNNs

There are many parameters in CNN, including biases, weights, the number of layers, neurons, activation function, stride, filter size, learning rate, and so on. Different correlation levels can be investigated using different filter sizes because convolutional operations consider the vicinity (locality) of input pixels. Different filter sizes encompass diverse levels of granularity; typically, fine-grained information is extracted by filters in small sizes, whereas coarse-grained information is extracted by filters in large sizes. As a result, in the early 2000s, spatial filters were used by investigators to increase performance. It was observed that there is relationship between a spatial filter and network learning. Various experiments undertaken during this time period indicated that, by adjusting filters, CNN could perform more effectively on coarse and fine-grained details.

2.2.2. CNN Based on Depth

The main idea behind deep CNN architecture is that, with the help of additional mappings (nonlinear) and more advanced feature hierarchies, the network can approximate the goal function effectively [9]. The network's depth has been an important parameter for supervised training. Deep networks represent specific function classes more effectively than shallow systems. In 2001, [10] proposed a theorem called "universal approximation". It explained that a single hidden layer may approximate any function. However, this happens at the cost of an exponentially enormous number of neurons and a computationally unrealistic result. Bengio and Delalleau postulated in 2011 that deeper networks can maintain the network's theatrical impact at a lesser cost [11,12]. Deep networks are computationally more efficient for complicated operations, according to Bengio, who demonstrated this empirically in 2013 [11]. VGG and Inception performed best in the ILSVRC-2014 competition, reinforcing the notion that depth is an important parameter in regulating the learning ability of networks [13–17].

2.2.3. CNNs with Multiple Paths

Deep CNNs are often good at complicated jobs. Sometimes they may suffer from performance degradation, explosion issues, or gradient disappearing, which are produced by increasing the depth rather than overfitting. The vanishing gradient problem leads to an increased test error and training error [18]. The theory of cross-layer connectivity or multi-path was proposed for deep training networks. Shortcut connections or numerous pathways can have a connection from one layer to another analytically by evading some in-between levels, allowing for the customized flow of information between the layers [19]. The network is split into the sections using cross-layer connectivity. These pathways solves the vanishing gradient problem by extending the gradient to lower layers.

2.2.4. Feature-Map Exploitation Based CNNs

CNN became popular for MV tasks due to its capacity to carry out hierarchical learning and automatic feature extraction [4]. The performance of classification, segmentation, and detection modules is heavily influenced by feature selection. CNN selects features dynamically by adjusting the weights associated with a kernel, also known as a mask. Furthermore, different feature extraction stages are performed, allowing for various types of features (known as feature maps or channels in CNN). However, some of the feature maps have little or no function in object discrimination [20]. Excessive feature sets may provide a noise effect, leading to the over-fitting of the network. This implies that, in addition to network engineering, the selection of feature maps can play a crucial role in increasing network generalization.

2.2.5. Multi-Connection Depending on the Width

During CNN advancements, the emphasis was mainly on leveraging the potential of the depth and the efficiency of connections in network regularization during 2012 to 2015. Kawaguchi, in 2019, discovered that the network width is equally essential [21].

This implies that, in addition to depth, width is an important component in developing learning philosophies. It is shown that neural networks with ReLU activation functions must be wide enough to retain a universal approximation property while also increasing in depth [22]. One significant issue with deep neural network architectures is that several layers may fail to learn valuable features. Although stacking many layers (raising depth) may learn varied feature representations, it does not always boost the NN's learning power. Furthermore, any deep network cannot arbitrarily approximate a class of continuous functions on a compact set if the network's maximum width is not greater than the input dimension [23]. Thus, the research focus switched from deep and narrow designs to wide and thin architectures to address this issue.

2.2.6. Exploitation-Based Feature-Map (ChannelFMap) CNNs

Because of its capacity to perform hierarchical learning and automatic feature extraction, CNN has received many interests in computer vision problems [4]. The performance of classification, segmentation, and detection modules is heavily influenced by feature selection. CNN selects features dynamically by adjusting the weights linked with a kernel, also known as a mask. In addition, many feature extraction phases are employed in CNN to mine various types of features. However, some feature maps have little or no significance in object discrimination. Massive feature sets may provide a noise effect, causing the network to overfit. This implies that, in addition to network engineering, the selection of feature maps can play an essential role in increasing network generalization. Feature maps and channel terms are frequently used interchangeably in the literature.

2.2.7. CNNs That Are Based on Attention

Diverse levels of abstraction play an essential role in determining the NN's discrimination power. Different hierarchies of abstractions focused on attributes relevant to picture localization and recognition play an essential role in learning. This effect is known as attention in the human visual system. Humans can view any scene by integrating partial glances of it and focusing on context-relevant aspects. This approach focuses on specified regions and comprehends numerous interpretations of items at a specific spot, hence improving visual structure capture. RNN and LSTM incorporate a more or less comparable interpretation. RNN and LSTM networks use attention modules as progressive feature, and the new tasters are weighted based on their recurrence in earlier rounds. The concept of attention in the convolutional neural network is used by various scholars to improve representation and overcome computational limitations. This concept of attention also contributes to CNN becoming intelligent enough to distinguish items even in busy backdrops and complex scenarios.

2.2.8. Dimension-Based CNN

The classic convolutions layer encodes both channel-wise and spatial information simultaneously, but it is computationally expensive. The efficiency of ordinary convolutions was enhanced by the introduction of separable (or depth-wise separable) convolutions [7], which encode spatial and channel-wise information separately using point-wise and depth-wise convolutions, respectively. This factorization is far more efficient, but it places a considerable computational burden on point-wise convolutions, making them a computational bottleneck. Table 1 shows summary of various CNN variants with its architectural categories.

Table 1. Performance summary of various CNN variants with its architectural categories.

Architecture Name	Year	Category	Main Role	Parameter	Error Rate
LeNet	1998	Spatial Exploitation	It was the first prevalent CNN architecture	0.060 M	dist]MNIST: 0.8 MNIST: 0.95
AlexNet	2012	Spatial Exploitation	• Deeper and wider compared to LeNet • Used RELU, dropout and overlap pooling GPUs NVIDIA GTX 580	60 M	ImageNet: 16.4
ZfNet	2014	Spatial Exploitation	Provided visualization of intermediate layers	60 M	ImageNet: 11.7
VGG	2014	Spatial Exploitation	It used small-sized kernels and had homogeneous topology	138 M	ImageNet: 7.3
GoogleNet	2015	Spatial Exploitation	It was first architecture to introduce block concept. It used split transform and then merge idea	4 M	ImageNet: 6.7
InceptionV-3	2015	Depth + Width	It was able to handle bottleneck issue and applied small filters rather than using large filters	23.6 M	ImageNet: 3.5 Multi-Crop: 3.58 Single-Crop: 5.6
Highway Network	2015	Depth + Multi-Path	First architecture to introduce the idea of multi path	2.3 M	CIFAR-10: 7.76
Inception V-4	2016	Depth + Width	It used asymmetric filters with split transform and merge concept	35 M	ImageNet: 4.01
Inception ResNet	2016	Depth + Width + Multi-Path	It used residual link with split transform and merge concept	55.8 M	ImageNet: 3.52
ResNet	2016	Depth + Multi-Path	Identified mapping-based skip connections with residual learning	25.6 M 1.7 M	ImageNet: 3.6 CIFAR-10: 6.43
Deluge Net	2016	Multi-path	Allowed cross layer information flow in deep networks	20.2 M	CIFAR-10: 3.76 CIFAR-100: 19.02
Fractal Net	2016	Multi-path	Various path lengths interacted with each other without any residual connection	38.6 M	CIFAR-10: 7.27 CIFAR-10+: 4.60 CIFAR-10++: 4.59 CIFAR-100: 28.20 CIFAR-100+: 22.49 CIFAR100++: 21.49
WideResNet	2016	Width	Width was increased in comparison to depth	36.5 M	CIFAR-10: 3.89 CIFAR-100: 18.85
Xception	2017	Width	Depth-based convolution was followed by point-based convolution	22.8 M	ImageNet: 0.055
Residual Attention Neural Network	2017	Attention	First architecture to introduce attention mechanism	8.6 M	CIFAR-10: 3.90 CIFAR-100: 20.4 ImageNet: 4.8
ResNext	2017	Width	Introduced cardinality, homogeneous topology and grouped convolution	68.1 M	CIFAR-10: 3.58 CIFAR-100: 17.31 ImageNet: 4.4
Squeeze and Excittation Network	2017	Feature-Map Exploitation	Modeled interdependencies between feature maps	27.5 M	ImageNet 2.3

Table 1. *Cont.*

Architecture Name	Year	Category	Main Role	Parameter	Error Rate
DenseNet	2017	Multi-Path	Crosslayer information flow	25.6 M 25.6 M 15.3 M 15.3 M	CIFAR-10+: 3.46 CIFAR100+: 17.18 CIFAR-10: 5.19 CIFAR-100: 19.64
PolyNet	2017	Width	Implemented structural diversity and poly-inception module and generalized residual unit	92 M	ImageNet: Single: 4.25 Multi: 3.45
PyramidalNet	2017	Width	Increased width gradually per unit	116.4 M 27.0 M 27.0 M	ImageNet: 4.7 CIFAR-10: 3.48 CIFAR-100: 17.01
Convolutional Block Attention Module (ResNeXt101 (32x4d) + CBAM)	2018	Attention	It exploited both spatial and feature map information	48.96 M	ImageNet: 5.59
Concurrent Spatial and Channel Excitation Mechanism	2018	Attention	Implemented spatial attention, feature-map attention, and concurrent placement of spatial and channel attention	-	MALC: 0.12 Visceral: 0.09
Channel Boosted CNN	2018	Channel Boosting	Boosted original channels with additional information by artificial channels	-	-
Competitive Squeeze and Excitation Network CMPE_SE_WRN_28	2018	Feature-Map Exploitation	Identity mapping and residual mapping were both used	36.92 M 36.90 M	CIFAR-10: 3.58 CIFAR-100: 18.47
EdgeNet	2019	Dimension Based	Introduced concept of visual intelligence at the edge.	-	-
ESPNetV2	2019	Dimension Based	Used light-weight and power-efficient general purpose CNN	-	68% accuracy
DiceNET	2021	Dimension Based	Introduced dimension-based CNN, including height, width, and depth	-	75.1% Accuracy

CNN Competitive Architectures

In the last five to seven years, around 367 papers showcased an architectural change in CNN as per dblp (computer science bibliography). These networks have become so large and deep that visualizing the complete model has become incredibly difficult. In the year 2000, there were around 100 papers, and 2021 had almost 60 papers until August 2021. This manuscript showcases some of the benchmarking CNN variants.

Mask R-CNN (2017)

This is a Faster R-CNN extension that improves a branch for forecasting an object mask simultaneously to the present bounding box branch identification. It is straightforward to train and adds only a minor amount of complexity. Overhead to a faster R-CNN running at five frames per second, it is easy to generalize other applications, such as human pose estimations, within the same framework [24].

G-RCNN (Graph Recognition Convolutional Neural Network) (2021)

This analyses a given image using deep CNN and turns the resulting information into a program code. It is a network that shares a rich convolutional feature vector calculation while simultaneously predicting edge and node information. A *flow chart* is a diagram

that depicts a program's workflow. It is commonly used in textbooks to teach coding and illustrate applications. Furthermore, flow charts are simple to use, allowing users to concentrate on programming ideas rather than language intricacies [25].

MFRNet: (2021) (Multi-Level Feature Review Network)

This is a unique CNN architecture for video compression by performing in-loop filtering and post-processing. It has four multi-level feature review residual dense blocks (MFRB) linked together in a cascade fashion. A multi-level residual learning framework and dense collections are used for each MFRB, which collects features from multiple convolutional layers. To optimize the information flow between these blocks even further, each reuses high-dimensional features provided by previous MFRB [26].

DDGD: Disentangled Dynamic Graph Deep Generation (2021)

This has demonstrated encouraging results in a variety of disciplines, including chemical design and protein structure prediction. Existing research, however, has mostly focused on static graphs. There is much less research carried out on dynamic graphs, which are important in protein folding, chemical reactions, and human movement applications. Encompassing existing deep generative models, from static to dynamic, is a difficult task that necessitates the factorization of static and dynamic features, as well as mutual interactions between node and edge patterns. This study develops a novel framework of factorized deep generative models for creating interpretable dynamic graphs [27].

YOLOv4 (2020)

The CNN accuracy is supposed to be improved through plenty of features. The practical testing of such feature combinations on large datasets is required, as is the theoretical justification of the results. Some aspects, such as batch normalization and residual connections, are only appropriate to specific models and issues or to small-scale datasets, but others are suitable to the broad majority of models, tasks, and datasets. To achieve cutting-edge results, this paper employs novel features, such as cross-mini batch-normalization, self-adversarial training, cross-stage partial connections, weighted residual connections, Mish activation, DropBlock regularization, and Mosaic data augmentation, as well as combining some of them [28].

Net2Vis (2021)

Appropriate visuals are critical for communicating neural network topologies in articles. While most contemporary deep learning publications include such visualizations, they are typically constructed shortly before publication, resulting in a lack of standard visual grammar, significant time investment, inaccuracies, and ambiguities. Current automatic network visualization techniques are designed to diagnose the network and are unsuitable for creating published visuals. As a result, they have provided a method for automating this process by converting Keras-specified network architectures into visualizations directly inserted into any publication [29].

Sketch-R2CNN (2021)

Sketches in today's large-scale datasets, such as the recently released QuickDraw collection, are commonly kept in vector format, with strokes composed of consecutively sampled points. In contrast, most recent sketch identification systems rasterize vector sketches as binary images before utilizing image classification techniques. This paper proposed a novel single branch network architecture to utilize vector of sketches for recognition [30].

DeepThin (2021)

For automated car driving applications, a powerful and accurate traffic sign detection system is necessary. This study created a novel energy-efficient thin but deep convolutional

neural network architecture for traffic sign recognition. It comprises fewer than 50 features in each convolutional layer, allowing CNN to be trained quickly, even without the assistance of a GPU [31].

YOLOX: Exceeding YOLO Series in 2021 (2021)

In this study, they offer some substantial improvements to the YOLO series, resulting in creating a new high-performance detector called YOLOX. To achieve state-of-the-art results over many models, they switch the YOLO detector to an anchor-free mode and use other sophisticated detection approaches, such as a decoupled head and the leading label assignment scheme SimOTA. For YOLO-Nano, with only 0.91 M parameters and 1.08 G FLOPs, they achieve 25.3 percent AP on COCO, outperforming NanoDet by 1.8 percent AP; for YOLOv3, one of the most widely used detectors in the industry, they achieve 47.3 percent AP on COCO, outperforming the current best practice by 3.0 percent AP; and for YOLOX-L, it has roughly the same number of parameters as YOLOv4. Furthermore, employing a single YOLOX-L model, they achieved first place in the Streaming Perception Challenge (Workshop on Autonomous Driving at CVPR 2021). This paper will benefit developers and researchers in real-world scenarios, who will also deploy versions that support ONNX, TensorRT, NCNN, and Openvino [32].

ChebNet: Chebyshev Polynomial Based Graph Convolution (2016)

ChebNet is regarded as one of the first and most influential papers on spectral graph learning. The multiplication of a signal (node features/attributes) by a kernel is a spectral convolution. It is analogous to how convolutions work on images, where a kernel value is multiplied by a pixel value. A spectral convolution kernel is composed by Chebyshev polynomials. Chebyshev polynomials are orthogonal polynomials with properties that make them excellent at tasks such as function approximation [33].

GCN: Graph Convolutional Network (2016)

Graphs can be found in a variety of application disciplines, including as bioinformatics, social analysis, and computer vision. The capacity of graphs to capture structural links among data allows for more insights than evaluating data in isolation. However, solving learning problems on graphs is generally difficult because (1) many forms of data, including as photos and text data, are not initially structured as graphs, and (2) the underlying connectivity patterns for graph-structured data are frequently complex and diverse. Representation learning, on the other hand, has found considerable success in a multitude of disciplines. As a result, learning how to represent graphs in a low-dimensional Euclidean space while preserving graph features is one feasible answer [34].

FastGCN: Minibatch Training for Graph Convolutional Network (2018)

The recently suggested GCN is an operational graph model for semi-supervised learning. Furthermore, because of the recursive neighborhood expansion across layers, training with large, dense graphs possesses time and memory issues. To avoid having test data accessible at the same time, this paper interprets graph convolutions as integral transformations of embedding functions. Thus, it uses Monte Carlo techniques to reliably guess the integrals, which leads to the batched training scheme [35].

LanczosNet (2019)

The Lanczos network (LanczosNet) is a graph convolution network that uses the Lanczos algorithm to generate low-rank approximations of the graph Laplacian. Using the Lanczos algorithm's tridiagonal decomposition, we not only efficiently exploit multiscale information via the quick approximated computation of the matrix power, but we also develop learnable spectral filters. LanczosNet, because it is fully differentiable, allows for both graph kernel learning and learning node embeddings. We demonstrate the relationship between our LanczosNet and graph-based manifold learning methods,

particularly diffusion maps. On citation networks and the QM8 quantum chemistry dataset, we compare our model to other recent deep graph networks. Experiment findings reveal that our model outperforms the competition in the majority of tasks [36].

SplineCNN (2018)

This is a deep neural network that processes an input that is irregularly structured and geometric, such as graphs or meshes. The fundamental contribution of the study is a unique B-spline-based convolution operator that, due to the local support property of B-spline basis functions, renders the calculation time independent of the kernel size. They generalize the classic CNN convolution operator by using continuous kernel functions with a fixed number of trainable weights. In contrast to comparable algorithms that filter in the spectrum domain, the suggested method aggregates characteristics only in the spatial domain. Furthermore, SplineCNN enables complete end-to-end deep architecture training through using simply the geometric structure as an input rather than handcrafted feature descriptors [37].

ECC: Edge-Conditioned Convolution (2017)

A prediction of graph-structured data can be used for a wide range of problems. In this study, they extended the convolution operator from normal grids to random graphs while avoiding the spectral domain, allowing them to handle graphs of various sizes and connectivity. Filter weights are conditioned on the specific edge labels in a vertex's proximity to go beyond essential diffusion. They examined the development of deep neural networks for graph classification in conjunction with the selection of the appropriate graph coarsening [38].

GAT: Graph Attention Network (2017) Ioffe 2013

These are novel architectures that operate on graph-structured data and employ masked self-attentional layers to address the shortcomings of previous systems that relied on graph convolutions or their approximations. For example, this paper allows (implicitly) for specifying different weights to different nodes in a neighborhood by stacking layers in which nodes can attend over the features of their neighborhoods without requiring costly matrix operations (such as inversion) or relying on prior knowledge of the graph structure. As a result, they address a number of critical challenges associated with spectral-based graph neural networks at the same time, and our model is easily adaptable to both inductive and transductive situations [39].

3. CNN Application

CNN is used for computer vision tasks, which solve image processing and ML-based problems, such as object identification, image recognition, image classification, image segmentation, and so on [40]. However, for training, CNN requires a huge amount of data. CNN has mostly demonstrated remarkable success in traffic sign identification, medical picture segmentation, face detection, and object identification in natural photos, where there is sufficient labeled data available for training. Figure 13. Shows various CNN applications.

3.1. Computer Vision and Associated Applications

Computer vision (CV) is the study of creating an artificial system that can perceive and extract usable information from visual data, such as photographs and movies. Face recognition, position estimation, activity recognition, and other applications are all covered by CV. Face recognition is one of the CV's most burdensome duties. Face recognition systems must deal with variances induced by lighting, changes in posture, and various face emotions. Deep CNN was proposed by Farfade in 2015 for face detection from various positions and occluded faces [41]. In 2016, Zhang used a new multitasking cascaded CNN for face detection in another study [42]. Compared to state-of-the-art approaches, Zhang's

technique yielded encouraging results. As there is considerable diversity in the body pose, "pose estimation in human" is the most challenging computer vision task. In 2016, Bulat and Tzimiropoulos proposed another cascade-based CNN approach [43]. The initial heat maps are recognized in their cascaded architecture, and then the regression is carried out on the detected heat maps in the second phase. One of the key aspects of activity detection is action recognition. While creating an action recognition system, there is a key difficulty in solving translations and distortions of features in diverse patterns that belong to the same action class. Earlier methods included creating motion history photographs, HMM (hidden Markov models), and action sketch generation. In 2017, Wang suggested a 3D CNN architecture paired with LSTM for identifying diverse activities from video frames [44]. Wang's technique surpasses other activity recognition-based algorithms in tests, according to the results [45]. Ji, in 2010, proposed another 3D CNN-based action recognition system. He used 3D CNN to extract information from many channels of input frames in his research [46]. On the combined extracted feature space, the final action recognition-based model is created. The proposed three-dimensional CNN model is supervised, trained, and can recognize activities in real-world scenarios.

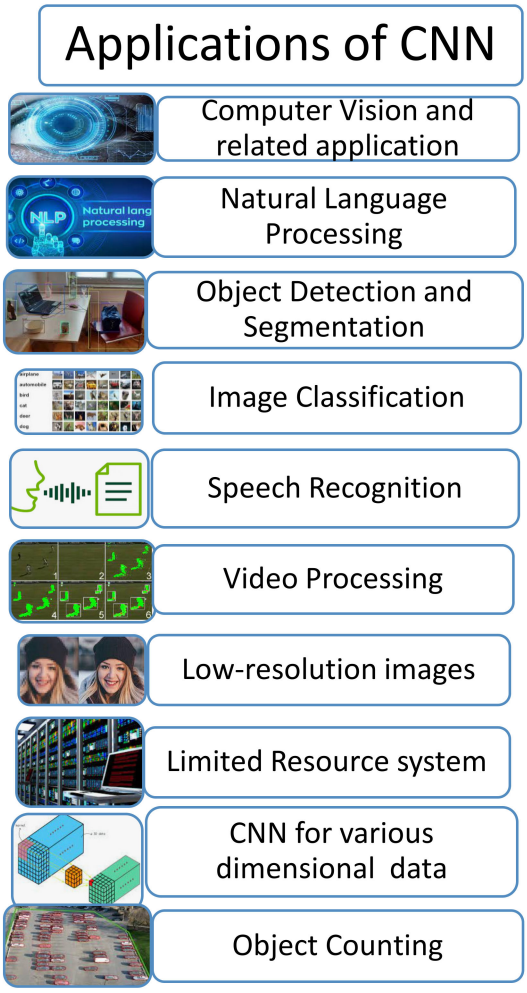


Figure 13. Applications of CNN.

3.2. Natural Language Processing (NLP)

The main task of natural language processing is to convert the language into a computer-friendly format. Although RNNs are well-suited to NLP applications, CNN has also been used in NLP-based applications, including language modeling and analysis. Since CNN was introduced as a new representation learning method, language modeling,

or sentence shaping, has taken a new turn. Sentence modeling is used to understand the semantics of sentences to create new and appealing applications that meet customers' needs. Data are analyzed using traditional information retrieval methods based on words or attributes, but the essence of the statement is ignored. During training, Kalchbrenner presented both a dynamic CNN and k-max pooling in 2014. This method discovers word relationships without relying on external sources, such as a parser or a dictionary [47]. Similarly, Collobert and Weston developed a CNN variant to execute multiple NLP-related tasks simultaneously, such as language modeling, chunking, named entity recognition, and semantic role modeling. In 2011, Hu proposed a general CNN variant that conducts sentence matching and can thus be used in various languages in another paper.

3.3. Object Detection and Segmentation

Object detection is concerned with identifying various items in photographs. R-CNN has recently become popular for object detection. In 2015, Ren proposed a rapid R-CNN for object detection as an enhancement over R-CNN (a fully connected convolutional neural network), which is employed in their research for feature extraction and is able to recognize the boundary and score of objects at various positions concurrently. Similarly, in 2016, Dai proposed employing fully connected CNNs to detect objects depending on their location. Gidaris et al. describe another object detection technique based on a multi-region-based deep CNN that aids in learning semantic aware features [48]. On the PASCAL VOC 2007 and 2012 datasets, Gidaris' method detects items with reasonable accuracy. AutoEncoder-based CNN architectures have recently demonstrated success in segmentation challenges. Various appealing CNN designs, including a fully convolutional network, SegNet, mask region-based CNN (R-CNN), U-Net, and others, have been described in this regard for both semantic and instance-based segmentation tasks [49].

3.4. Image Classification

CNN is widely used for image classification tasks. Medical images are one of CNN's critical uses, particularly for cancer diagnosis and utilizing histological images. Ref. [50] employed CNN to diagnose breast cancer photos and compared the results to a pre-trained network with a dataset using handmade descriptors. To deal with the problem of class skewness, data augmentation is used in the second phase. There are several popular image classification pre-trained networks available. Image classification can be easy if a labelled dataset can be produced for the target image.

3.5. Recognition of Speech

CNN is often regarded as the most effective method to deal with image processing tasks. However, recent research has revealed that it is also capable of performing well for speech recognition tasks. Hamid, in 2012, disclosed a speaker-independent voice recognition system using CNN [51]. In comparison to previously published methodologies, experimental results revealed a ten percent reduction in the mistake rate. Furthermore, after establishing the network, the performance of CNN is tested utilizing the pre-training phase. Experiments revealed that almost all of the architectures investigated perform well in the vocabulary and phone recognition tasks. CNN is becoming recognized for speech emotion recognition these days. For identifying speech emotions, Huang et al. employed CNN in conjunction with LSTM [52]. CNN was trained on both verbal and nonverbal portions of speech in Huang's technique, and CNN learned characteristics that were employed by LSTM to recognize speech emotions.

3.6. Video Processing

Video processing algorithms use the temporal and spatial information of videos. Many researchers have employed CNN to solve challenges connected to video processing [53]. For example, a straightforward border detecting method based on CNN was suggested. TAGs are created using CNN in Tong's method [19]. During the experiment, TAGs are

merged against a single shot to annotate that particular video. In 2016, Wang used 3D CNN and LSTM to distinguish activity in the video. In 2016, Frizzi employed CNN in another technique to identify some emergencies, such as fire or smoke, in the video [54]. According to Frizzi, CNN architecture is able to extract prominent characteristics and also perform the classification task. However, the collecting of geographical and temporal information is a time-consuming operation in action recognition. Shi Y, in 2017, presented a three-stream-based structure to solve the shortcomings of existing feature descriptors [55]. This structure is capable of extracting spatial–temporal characteristics as well as short and long-term motion inside the video. CNN uses bi-directional LSTM to recognize activity in the video, as stated in the paper of [56]. Their strategy is divided into two parts. First, the sixth frame of the videos is used to extract features in the first phase. Then, the bi-directional LSTM framework is used in the second phase to utilize sequential information between frame features.

3.7. Images with Low Resolution

Different researchers in ML have employed CNN-based image enhancement approaches to improve image resolution [57–59]. Peng et al. employed a deep CNN-based technique to identify items in low-resolution pictures [58]. LR-CNN was introduced by Chevalier et al. for low-resolution picture categorization [57]. Kawashima et al. describe another deep-learning-based technique in which convolutional layers and an LSTM layer are used to discern action from low-resolution thermal pictures [59].

3.8. System with Limited Resources

In 2017, Bettoni built CNN on top of the FPGA architecture to deal with power efficiency and mobility for embedded devices. Despite its high processing cost, CNN has been effectively used to develop several machine-learning-based embedded devices. For example, the number plate recognition system, created by Lee et al. is able to instantly recognize the number written on the license plate [60]. This embedded recognition system is based on a deep learning approach, which is based on a simple AlexNet architecture. Another solution uses the FPGA embedded platform to efficiently perform various CNN-based machine learning tasks [61]. Similarly, CNN designs with fewer resources, such as MobileNet, ShuffleNet, ANTNets, and others, are ideal for mobile devices [62]. Researchers have combined the MobileNet architecture with SSD to use MobileNet's lightweight architecture, which can be quickly installed on resource-constrained hardware and can learn enhanced representations from incoming [63].

3.9. CNN for Various Dimensional Data

Not only has CNN performed well on images, but it has also performed well on 1D data. As a result of its high feature extraction ability, 1D-CNN is becoming more popular than other ML methods. For intrusion detection in network traffic, Vinayakumar, in 2017, used 1D-CNN in conjunction with LSTM, RNN, and gated recurrent units [64]. Vinaykumar and co-researchers tested the proposed models and found that CNN outperforms classical ML models by a large margin. They developed an end-to-end system that can automatically extract damage-sensitive features from accelerated signals for detection purposes. Similarly, Yildirim et al. showed the successful use of CNN for the 1D biomedical dataset [65]. Three-dimensional shape models are becoming more readily available and easier to collect, making 3D data essential for item classification advancement. To overcome this challenge, current state-of-the-art approaches rely on CNN. CNN based on volumetric representations and CNN based on multi-view representations are two forms of CNN that have recently been developed. Existing volumetric CNN architectures and techniques cannot completely harness the power of 3D representations, as evidenced by empirical results from these two types of CNN. According to a comprehensive review of existing techniques, this work tries to improve both the volumetric CNN and multi-view CNN. To that purpose, two distinct network architectures of volumetric CNN are introduced. In addition, we look into

multi-view CNN, where multi-resolution filtering is used in 3D. Overall, both volumetric CNN and multi-view CNN outperform existing state-of-the-art approaches. Extensive experiments are provided to test underlying design decisions, allowing us to better grasp the space of object categorization algorithms possible for 3D data.

3.10. Object Counting

One of the essential jobs in computer vision is counting objects in images. This has a wide range of applications, including microbiology (e.g., counting bacterial colonies in a Petri dish), surveillance (e.g., counting people), agriculture (e.g., counting fruits or vegetables), medicine (e.g., counting tumor cells in histopathological images), and wildlife conservation (e.g., counting animals). Counting things is a simple task for humans, but it can be difficult for computers. Pre-trained YOLO can count the number of objects by removing the bottom prediction layer and feeding the characteristics to a classification feed-forward layer.

4. CNN Challenges

Deep CNNs have shown to be effective for data that are either time-series or have a grid-like structure. Deep CNN architectures have also been used to solve several additional problems.

Different researchers have enthralling arguments about the performance of CNN on various ML tasks. The following are some of the difficulties encountered while training deep CNN models:

- As deep CNNs are typically a black box, they may be challenging to comprehend and explain. As a result, verifying them can be challenging at times;
- According to Szegedy et al. (2013), training a CNN on noisy picture data can increase the misclassification error (Szegedy et al., 2014). Adding a small amount of random noise to the input image can deceive the network, causing the original and slightly agitated variant to be classified incorrectly;
- Each CNN layer is organized in such a manner that it extracts problem-specific information associated with the task automatically. In some of the cases, before classification, some jobs require knowledge of the behavior of features retrieved by deep CNN. Thus, the feature visualization concept in CNN may be helpful. Similarly, Hinton stated that lower levels should only pass on their knowledge to the relevant neurons of the following layer. Hinton presented an intriguing capsule network technique in this area [66];
- Deep CNNs use supervised learning processes, which require huge amount of annotated data to train the network. Humans, on the other hand, can learn from a few examples;
- The choice of hyper-parameters has a significant impact on the CNN performance. A slight change can influence the overall performance of a CNN in hyper-parameter values. As a result, selecting hyper-parameters with care is a critical design issue that must be addressed using an appropriate optimization technique;
- Effective CNN training necessitates the use of robust hardware resources, such as GPUs. However, the effective use of CNNs in embedded and intelligent devices is still required [53,67];
- One of CNN's limitations in vision-related jobs is that it rarely performs well when used to estimate an object's pose, orientation, or location. In 2012, AlexNet attempted to address this difficulty by developing data augmentation, which solved the problem to some extent. In addition, data augmentation aids CNN in learning a variety of internal representations, potentially improving its performance.

Spatial exploitation: Since convolutional operations take into account the neighborhood (correlation) of input pixels, different levels of the correlation can be examined by utilizing different filter sizes. Table 2 describes major challenges in spatial exploitation based CNN architecture.

Table 2. Major issues linked with the deployment of CNN architectures based on spatial exploitation.

Architecture	Merits	Demerits
LeNet	• Used spatial correlation to minimize computation and parameter count; • Feature hierarchies were automatically learned.	• Poor scaling to various picture classes; • Large size filters; • Low level feature extraction.
AlexNet	• Extraction of low, mid, and high-level features utilizing large and small size filters on the initial (5×5 and 11×11) and final layers (3×3); • Introduced regularization in CNN to give a notion of deep and extensive CNN architecture; • Began using GPUs as an accelerator in parallel to deal with complex architectures.	• Neurons in the first and second layers that are dormant; • Aliasing artefacts in trained feature-maps as a result of excessive filter size.
ZfNet	• Demonstrated parameter adjustment by seeing the output of intermediary layers; • Reduced the filter size and stride in AlexNet's first two layers.	• Visualization needs additional information processing.
VGG	• Proposed the concept of an effective receptive field; • Introduced the concept of simple and homogeneous topology.	• The use of computationally costly fully linked layers.
GoogleNet	• Presented the concept of utilizing multi-scale filters within the layers; • Introduced the concept of divide, transform, and merge; • Introduced the concept of divide, transform, and merge; • Used auxiliary classifiers to boost convergence rate.	• Difficult parameter tuning owing to heterogeneous topology; • Due to a representational bottleneck, useful information may be lost.

Depth-Based: The network can better approximate the target function using a number of nonlinear mappings and enhanced feature representations as its depth increases. The main hurdle that deep architectures face is the issue of vanishing gradients and negative learning. Table 3 describes major challenges in depth based CNN architecture.

Table 3. Major issues linked with the deployment of depth-based CNN architectures.

Architecture	Merits	Demerits
Inception-V3	• Asymmetric filters and a bottleneck layer were used to reduce the computational cost of deep systems.	• Architecture design is complex; • Deficiency of homogeneity.
Highway Networks	• Added a training method for deep neural networks; • Auxiliary connections were used in addition to direct connections.	• Difficult to implement parametric gating mechanism.
Inception-ResNet	• The power of residual learning and the inception block have been combined.	
Inception-V4	• Deep feature hierarchies, multilevel feature representation.	• Learning is slow.
ResNet	• Reduced the error rate for deeper networks; • Introduced the concept of residual learning; • Mitigated the effect of the vanishing gradient problem.	• A little complicated architecture; • Degraded feature-map information in feed forwarding; • Over-adaption of hyper-parameters for specific job owing to module stacking.

Multi-Path: Shortcut paths provide the option to skip some layers. Different types of shortcut connections used in literature are zero padded, projection, dropout, 1×1 connections, etc. Table 4 describes major challenges in multi path based CNN architecture.

Table 4. Major issues linked with the deployment of multi-path-based CNN architectures.

Architecture	Merits	Demerits
Highway Networks	By introducing cross-layer connectivity, it mitigates the constraints of deep networks.	Because gates are data dependent, they may become expensive.
Resent	• Use of identity-based skip connections to provide cross-layer connectivity; • Data independence and parameter-free information flow gates; • Signal can be readily passed in both directions, forward and backward.	• Many layers may offer very little or no information; • Redundant feature-maps may be re-learned.
DenseNet	• Introduced depth or cross-layer dimension; • Ensures maximum data flow between network layers; • Avoids relearning of redundant feature-maps; • Both low and high level features are accessible to decision layers.	• Significant parameter increase due to an increase in the number of feature-maps at each layer.

Width-Based: Previously, it was considered that increasing the number of layers would improve the accuracy. However, when the number of layers increases, the vanishing gradient problem develops, and training may become slow. As a result, the concept of layer widening was also examined. Table 5 describes major challenges in width based CNN architecture.

Table 5. Major issues linked with the deployment of width-based CNN architectures.

Architecture	Merits	Demerits
Wide ResNet	• Demonstrates the usefulness of parallel transformation utilization by expanding the breadth of ResNet while decreasing its depth; • Allows for feature reuse; • Has demonstrated that dropouts between the convolutional layer are more effective.	• There is a possibility of over-fitting; • There are more parameters than in thin deep networks.
Pyramidal Net	• Introduced the concept of progressively increasing the width each unit; • Avoids rapid information loss; • Covers all feasible locations rather than retaining the same dimension until the last unit.	• High spatial and time complexity; • May become quite complex, if layers are substantially increased.
Xception	• Introduced the concept that learning filters in 2D followed by 1D are easier than learning filters in 3D space; • Introduced depth-wise separable convolution; • Uses cardinality to discover good abstractions.	• Computational cost is high.
Inception	• Using varying size filters within the inception module boosts the output of the intermediate layers; • Using different size filters might help you to capture the variation in high-detail photographs.	• Space and time complexity will increase.
ResNeXt	• Added cardinality to provide various transformations at each layer; • Homogeneous topology allows for easy parameter customization.	• Computational cost is high.

Feature-Map Selection: As the deep learning topology is extended, an increasing amount of features maps are generated at each step. Many of the feature-maps might be important for the classification task, whereas others might be redundant or less important. Hence, feature-map selection is another important dimension in deep learning architectures. Table 6 describes major challenges in feature map based CNN architecture.

Table 6. Major issues linked with the deployment of feature-map-based CNN architectures.

Architecture	Merits	Demerits
Squeeze and Excitation Network	• It is a block-based concept; • It introduced a generic block that, due to its simplicity, can be simply incorporated to any CNN model • It squeezes fewer important characteristics, and vice versa.	• In ResNet, the weight of each channel is determined only by the residual information.
Competitive Squeeze and Excitation Networks	• Makes use of feature-map statistics derived from both residual and identity mapping-based features; • Puts residual and identification feature-maps to the test.	• Doesn't accompany the concept of attention.

Channel Boosting: CNN learning is also dependent on the input representation. The CNN performance may be hampered by a lack of diversity and class discernible information in the input. To that end, the notion of channel boosting (input channel dimension) utilizing auxiliary learners is introduced in CNN to improve the network's representation [1]. Table 7 describes major challenges in channel-boosting based CNN architecture.

Table 7. Major issues linked with the deployment of channel-boosting-based CNN architectures.

Architecture	Merits	Demerits
Channel Boosted CNN using Transfer Learning	• It increases the number of input channels to improve the network's representational capacity; • Inductive transfer learning is employed in an innovative approach to generate a boosted input representation for CNN.	• Increases in computational burden may occur as a result of the creation of auxiliary channels.

Attention-Based: There are many advantages of attention networks in determining which patch is the focus or most essential in a picture. Table 8 describes major challenges in attention based CNN architecture.

Table 8. Major issues linked with the deployment of attention-based CNN architectures.

Architecture	Merits	Demerits
Residual Attention Neural Network	• Creates attention-aware feature-maps that are easy to scale up because of residual learning; • Provides distinct representations of the targeted patches; • Adds soft weights to features using bottom-up top-down feed-forward attention.	• High complexity in model.
Convolutional Block Attention Module	• CBAM is a generic building block for feed-forward convolutional neural networks; • Produces both a feature-map and spatial attention in a sequential fashion. • Channel attention maps assist in determining where to focus one's attention; • Spatial attention aids in determining where to focus; • Improves the flow of information; • Employs both global average pooling and maximum pooling at the same time.	• There are chances of a high computational load.

Dimension-Based: Dimension-wise convolutions use light-weight convolutional filtering across each dimension of the input tensor, whereas dimension-wise fusion merges these dimension-wise representations efficiently. Table 9 describes major challenges in dimension based CNN architecture.

Table 9. Major issues linked with the deployment of dimension-based CNN architectures.

Architecture	Merits	Demerits
Dice-Net	• It includes two main novel aspects: dimension-wise convolutions and dimension-wise fusion; • Dimension-wise convolutions use light-weight convolutional filtering on each dimension of the input tensor; • Dimension-wise fusion mixes these dimension-wise representations efficiently; • High accuracy in image recognition.	• It may increase the time for producing results.

5. Future Directions

The use of numerous novel concepts in CNN's architecture has shifted research priorities, particularly in the field of computer vision. To study innovations in CNN's architecture is an encouraging study area, and has the potential to become one of the utmost utilized AI techniques.

- Ensemble learning is an upcoming research area in CNN. By extracting distinct semantic representations, the model can improve the generalization and resilience of many categories of images by combining multiple and diverse designs;
- In picture segmentation tasks, although it performs well, a CNN's ability as a "generative learner" is limited. The use of CNNs' generative learning capabilities throughout feature extraction phases can improve the model's representational power. At the intermediate phases of CNN, fresh examples can be incorporated to improve the learning capability by using auxiliary learners (Khan et al., 2018a);
- Attention is a crucial process in the human visual system for acquiring information from images. Furthermore, the attention mechanism collects the crucial information from the image and stores its context in relation to the other visual components. In the future, the research could be conducted to preserve the spatial importance of objects and their distinguishing characteristics during subsequent stages of learning;
- It is observed that the learning capability of a CNN is mainly increased by increasing the network's size, and this may be achieved by modern advanced hardware technologies, such as the Nvidia DGX-2 supercomputer. Nonetheless, training more deep and high-capacity CNN architectures consumes a substantial amount of memory and computing resources [68,69];
- The fundamental disadvantage of CNNs is their inability to be applied in real-time. Furthermore, the CNN is delayed in compact hardware due to its high computational cost, particularly in mobile systems. Therefore, various hardware accelerators are necessary for this scenario to reduce the execution time and power consumption. Thus far, numerous highly interesting accelerators have been presented in this field. Examples include Eyeriss, FPGA, and application-specific integrated circuits (Moons and Verhelst 2017);
- The activation function (e.g., RELU, sigmoid, etc.), number of neurons per layer, kernel size, layer organization, and other hyper-parameters of deep CNN are critical. There is a trade-off between the selection of hyper-parameters and the evaluation time. Hyper-parameter tuning is a time-consuming and intuitive process that cannot be specified explicitly. In this case, genetic algorithms can be used to automatically

enhance hyper-parameters by conducting searches both at random and by directing searches based on previous results (Khan et al., 2019);

- Deep and broad CNN poses a significant difficulty in developing and executing them on devices with limited resources;
- Pipeline parallelism can be utilized to scale up in-depth CNN training to overcome hardware limitations. The Google group has presented GPipe, a distributed machine learning library that includes a model parallelism option for training. Pipelining could be utilized in the future to speed up the training of big models and scale performance without having to tune hyper parameters;
- Cloud-based platforms' promise in creating computationally expensive CNN applications is projected to be fully realized in the future. Cloud computing helps the user to deal with large amounts of data and provides them with an exceptional computational efficiency at a reasonable cost. Amazon, Microsoft, Google, and IBM, among others, provide public cloud computing resources with superb scalability, speed, and flexibility for training-resource-intensive CNN designs. Furthermore, the cloud environment makes it simple for researchers and new practitioners to set up libraries;
- As CNN primarily uses image processing, implementing state-of-the-art CNN architectures on sequential data necessitates transforming 1D data to 2D data. The trend of using 1D-CNNs for sequential data is being advocated because of their excellent feature extraction capabilities and efficient computations with a small number of parameters [70];
- High-energy researchers at CERN have recently used CNN's learning capabilities to investigate particle collisions. The use of machine learning, specifically deep CNN, in high-energy physics is projected to increase [70,71].
- Human activity recognition is trending research area in the field of CNN. References [72,73] have described the various CNN variants for human activity and pose recognition.

6. Conclusions

In the recent decade, convolutional neural networks have received much attention. They have a large impact on image processing and vision-related tasks, which has piqued academics' curiosity. Many academics have carried out outstanding work in this area, modifying the CNN design to improve its performance. Changes in activation functions, developing or modifying loss functions, optimization, architectural innovations, application-specific modifications in architecture, developing various learning algorithms, and regularization are some of the categories in which researchers have made advancements in CNN. This manuscript summarizes recent developments in CNN architectures. The eight fundamental architectural advances in CNN are spatial exploitation, depth, multi-path, breadth, dimension, feature-map exploitation, channel boosting, and attention. It can be concluded by surveying various architectural modifications in CNN that CNN's block-based architecture supports modular learning, making the architecture more basic and accessible. Another dimension-based category has a positive impact on CNN's total performance. Dimension-based CNN can also be used to recognize three-dimensional objects. Training CNN for an exemplary performance in 3D object recognition is a promising and complicated research field. Researchers in this area can still work on 3D object recognition and NAS-based techniques. Modular or block-based architecture has shown excellent optimization in both time and accuracy.

Author Contributions: Conceptualization, D.B. and C.P.; methodology, D.B.; software, H.T.; validation, J.P., R.V. and K.M.; formal analysis, S.P.; investigation, H.G.; resources, D.B.; data curation, C.P.; writing—original draft preparation, D.B.; writing—review and editing, C.P.; visualization, H.T.; supervision, K.M.; project administration, D.B.; funding acquisition, C.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Acknowledgments: The authors would like to thank the reviewers for their valuable suggestions which helped in improving the quality of this paper.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Khan, A.; Sohail, A.; Zahoora, U.; Qureshi, A.S. A Survey of the Recent Architectures of Deep Convolutional Neural Networks. *Artif. Intell. Rev.* **2020**, *53*, 5455–5516. [CrossRef]
2. Liu, X. Recent progress in semantic image segmentation. *Artifical Intell. Rev.* **2019**, *52*, 1089–1106. [CrossRef]
3. Deng, L.; Dong, Y. Deep Learning: Methods and Applications. In *Foundations and Trends R in Signal Process*; Now Publishers Inc.: Boston, MA, USA, 2013.
4. LeCun, Y. Convolutional networks and applications. *ISCAS IEEE* **2010**, 253–256. [CrossRef]
5. Najafabadi, M.M. Deep learning applications and. *J. Big Data* **2015**, *2*, 1–21. [CrossRef]
6. Guo, Y. Deep learning for visual understanding: A review. *Neurocomupting* **2016**, *187*, 27–48. [CrossRef]
7. Towards Datascience. Available online: https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6 (accessed on 27 July 2021).
8. Towards Datascience. Available online: https://towardsdatascience.com/convolutional-neural-networks-explained-9cc5188c4939 (accessed on 29 July 2021).
9. Bengio, Y. Deep learning of representations: Looking forward. In Proceedings of the International Conference on Statistical Language and Speech Processing, Tarragona, Spain, 29–31 July 2013; Springer: Berlin/Heidelberg, Germany, 2013.
10. Balázs, C.C. Approximation with Artificial Neural Networks. Master's Thesis, Eötvös Loránd University, Budapest, Hungary, 2001.
11. Delalleau, O. Shallow vs. deep sum-product networks. *Adv. Neural Inf. Process. Syst.* **2011**, 666–674. [CrossRef]
12. Szegedy, C. Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning. *arXiv* **2016**, arXiv:160207261v2.
13. Ioffe, S. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. *arXiv* **2015**, arXiv:1502.03167.
14. Szegedy, C. Rethinking the Inception Architecture for Computer Vision. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2015; IEEE: New York, NY, USA, 2016.
15. Szegedy, C. Going deeper with convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; IEEE: New York, NY, USA, 2015.
16. Simonyan, K. Very deep convolutional networks for large-scale image recognition. *ILCR* **2014**, *75*, 398–406.
17. Dong, C. Image super-resolution using deep convolutional networks. *IEEE Trans. Pattern Anal. Mach Intell.* **2016**, *38*, 295–307. [CrossRef]
18. Tong, T. Image super-resolution using dense skip connections. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017.
19. Hu, J. Squeeze-and-Excitation Networks. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018.
20. Kawaguchi, K. Effect of depth and width on local minima in deep learning. *Neural Comput.* **2019**, *31*, 1462–1498. [CrossRef]
21. Hanin, B. Approximating Continuous Functions by ReLU Nets of Minimal width. *arXiv* **2017**, arXiv:171011278.
22. Nguyen, Q. Neural Networks Should Be Wide Enough to Learn Disconnected Decision Regions. *arXiv* **2018**, arXiv:180300094.
23. He, K. Mask R-CNN. *arXiv* **2018**, arXiv:1703.06870.
24. Lin, C. GRCNN: Graph Recognition Convolutional Neural Network for Synthesizing Programs from Flow Charts. *arXiv* **2020**, arXiv:2011.05980.
25. Ma, D. MFRNet: A New CNN Architecture for Post-Processing and In-loop Filtering. *arXiv* **2020**, arXiv:2007.07099v2.
26. Zhang, W. Disentangled Dynamic Graph Deep Generation. *arXiv* **2021**, arXiv:2010.07276v2.
27. Alexey, B. YOLOv4: Optimal Speed and Accuracy of Object Detection. *arXiv* **2020**, arXiv:2004.10934v1.
28. Aex, B. Net2Vis—A Visual Grammar for Automatically Generating Publication-Tailored CNN Architecture Visualizations. *arXiv* **2021**, arXiv:1902.04394v6.
29. Zou, C.; Zheng, Y.; Su, Q.; Fu, H. Chiew-Lan Tai Sketch-R2CNN: An Attentive Network for Vector Sketch Recognition. *arXiv* **2018**, arXiv:1811.08170v1.
30. Haque, W.A. DeepThin: A novel lightweight CNN architecture for traffic sign recognition without GPU requirements. In *Expert Systems with Applications*; Elseveir: Amsterdam, The Netherlands, 2021; Volume 168.
31. Zheng, G. YOLOX: Exceeding YOLO Series in 2021. *arXiv* **2021**, arXiv:2107.08430v2.
32. Defferrard, M. Convolutional neural networks on graphs with fast localized spectral filtering. *Adv. Neural Inf. Process. Syst.* **2016**, 3844–3852.
33. Kipf, T.N. Semi-supervised classification with graph convolutional networks. *arXiv* **2016**, arXiv:1609.02907.
34. Chen, J. Fastgcn: Fast learning with graph convolutional networks via importance sampling. *arXiv* **2018**, arXiv:1801.10247.
35. Liao, R. Lanczosnet: Multiscale deep graph convolutional networks. *arXiv* **2019**, arXiv:1901.01484.
36. Fey, M. Splinecnn: Ffast geometric deep learning with continuous b-spline kernels. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–26 June 2018; pp. 869–877.

37. Simonovsky, M. Dynamic edge-conditioned filters in convolutional neural networks on graphs. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 June 2017; pp. 3693–3702.
38. Velickovic, P. Graph attention networks. *arXiv* **2017**, arXiv:1710.10903.
39. Chouhan, N.; Khan, A.; Khan, H.-R. Network anomaly detection using channel boosted and residual learning based deep convolutional neural network. *Appl. Soft Comput.* **2019**, *83*, 105612. [CrossRef]
40. Farfade, S.S. Multi-view Face Detection Using Deep Convolutional Neural Network. In Proceedings of the 5th ACM on International Conference on Multimedia Retrieval—ICMR '15, Shanghai, China, 23–26 June 2015; ACM Press: New York, NY, USA, 2015; pp. 643–650.
41. Zhang, K. Joint face detection and alignment using multitask cascaded convolutional networks. *IeeexploreIeeeOrg* **2016**, *23*, 1499–1503. [CrossRef]
42. Bulat, A.; Tzimiropoulos, G. Human Pose Estimation via Convolutional Part Heatmap Regression BT. In *Computer Vision—ECCV 2016*; Leibe, B., Matas, J., Sebe, N., Welling, M., Eds.; Springer: Berlin/Heidelberg, Germany, 2016; pp. 717–732.
43. Wang, X. Beyond Frame-level CNN: Saliency-Aware 3-D CNN With LSTM for Video Action Recognition. *IEEE Signal Process. Lett.* **2016**, *24*, 510–514. [CrossRef]
44. Wang, H.; Schmid, C. Action recognition with improved trajectories. In Proceedings of the IEEE International Conference on Computer Vision, Sydney, Australia, 1–8 December 2013; pp. 3551–3558.
45. Ji, S.; Xu, W.; Yang, M.; Yu, K. 3D convolutional neural networks for human action recognition. *ICML Int. Conf. Mach. Learn.* **2010**, *35*, 221–231. [CrossRef]
46. Kalchbrenner, N.; Grefenstette, E.; Blunsom, P. A convolutional neural network for modelling sentences. *arXiv* **2014**, arXiv:14042188.
47. Gidaris, S.; Komodakis, N. Object detection via a multi-region and semantic segmentation aware U model. In Proceedings of the IEEE International Conference On Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1134–1142.
48. Kendall, A.; Cipolla, R.; Badrinarayanan, V. SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 2481–2495.
49. Spanhol, F.A.; Oliveira, L.S.; Petitjean, C.; Heutte, L. A dataset for breast cancer histopathological image classification. *IEEE Trans. Biomed. Eng.* **2016**, *63*, 1455–1462. [CrossRef]
50. Abdel-Hamid, O. Applying convolutional neural networks concepts to hybrid NN-HMM model for speech recognition. In Proceedings of the 2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Kyoto, Japan, 25–30 March 2012; pp. 4277–4280.
51. Huang, K.Y. Speech Emotion Recognition Using Deep Neural Network Considering Verbal and Nonverbal Speech Sounds. In Proceedings of the ICASSP 2019–2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Brighton, UK, 12–17 May 2019.
52. Lu, Z. The expressive power of neural networks: A view from the width. In Proceedings of the 31st International Conference on Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; pp. 6231–6239.
53. Frizzi, S. Convolutional neural network for video fire and smoke detection. In Proceedings of the IECON 2016-42nd Annual Conference of the IEEE Industrial Electronics Society, Florence, Italy, 23–26 October 2016; pp. 877–882.
54. Shi, Y. Sequential deep trajectory descriptor for action recognition with three-stream CNN. *IEEE Trans. Multimed.* **2017**, *19*, 1510–1520. [CrossRef]
55. Ullah, A. Action recognition in video sequences using deep bi-directional LSTM with CNN features. *IEEE Access* **2017**, *6*, 1155–1166. [CrossRef]
56. Chevalier, M. LR-CNN for fine-grained classification with varying resolution. In Proceedings of the 2015 IEEE International Conference on Image Processing (ICIP), Quebec City, QC, Canada, 27–30 September 2015; pp. 3101–3105.
57. Peng, X. Fine-to-coarse knowledge transfer for low-res image classification. In Proceedings of the 2016 IEEE International Conference on Image Processing (ICIP), Phoenix, AZ, USA, 25–28 September 2016; pp. 3683–3687.
58. Kawashima, T. Action recognition from extremely low-resolution thermal image sequence. In Proceedings of the 2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance, AVSS 2017, Lecce, Italy, 29 August–1 September 2017; pp. 1–6.
59. Lee, S. Car Plate Recognition Based on CNN Using Embedded System with GPU. In Proceedings of the 2017 10th International Conference on Human System Interactions (HSI), Ulsan, Korea, 17–19 July 2017; pp. 239–241.
60. Xie, W. An Energy-Efficient FPGA-Based Embedded System for CNN Application. In Proceedings of the IEEE International Conference on Electron Devices and Solid State Circuits (EDSSC), Shenzhen, China, 6–8 June 2018; pp. 1–2.
61. Zhang, X. ShuffleNet: An Extremely Efficient Convolutional Neural Network for Mobile Devices. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018.
62. Shakeel, M.F. Detecting Driver Drowsiness in Real Time through Deep Learning Based Object Detection. In *Lecture Notes in Computer Science in Artificial Intelligence and Bioinformatics*; Springer: Berlin/Heidelberg, Germany, 2019.
63. Vinayakumar, R. Applying convolutional neural network for network intrusion detection. In Proceedings of the 2017 International Conference on Advances in Computing, Communications and Informatics, ICACCI 2017, Udupi, India, 13–16 September 2017.
64. Yıldırım, Ö. Arrhythmia detection using deep convolutional neural network with long duration ECG signals. *Comput. Biol. Med.* **2018**, *102*, 411–420. [CrossRef]

65. De Vries, H. Deep learning vector quantization. In Proceedings of the European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning, Bruges, Belgium, 27–29 April 2016.
66. Hinton, G. Matrix capsules with EM routing. In Proceedings of the 6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, 30 April–3 May 2018.
67. Justus, D. Predicting the Computational Cost of Deep Learning Models. In Proceedings of the 2018 IEEE International Conference on Big Data Big Data, Seattle, WA, USA, 10–13 December 2018.
68. Sze, V. *Efficient Processing of Deep Neural Networks: A Tutorial and Survey*; IEEE: Piscataway, NJ, USA, 2017.
69. Madrazo, C.F. Application of a Convolutional Neural Network for image classification for the analysis of collisions in High Energy. *EPJ Web Conf.* **2019**, *214*, 06017. [CrossRef]
70. Aurisano, A. A convolutional neural network neutrino event classifier. *J. Instrum.* **2016**, *11*, P09001. [CrossRef]
71. Liu, W. A survey of deep neural network architectures and their applications. *Neurocomputing* **2017**, *234*, 11–26. [CrossRef]
72. Patel, C.I.; Labana, D.; Pandya, S.; Modi, K.; Ghayvat, H.; Awais, M. Histogram of Oriented Gradient-Based Fusion of Features for Human Action Recognition in Action Video Sequences. *Sensors* **2020**, *20*, 7299. [CrossRef]
73. Patel, C.I.; Garg, S.; Zaveri, T.; Banerjee, A.; Patel, R. Human action recognition using fusion of features for unconstrained video sequences. *Comput. Electr. Eng.* **2018**, *70*, 284–301. [CrossRef]