Artificial Intelligence Review
https://doi.org/10.1007/s10462-020-09825-6

A survey of the recent architectures of deep convolutional neural networks

Asifullah Khan[1,2] · Anabia Sohail[1,2] · Umme Zahoora[1] · Aqsa Saeed Qureshi[1]

Abstract
Deep Convolutional Neural Network (CNN) is a special type of Neural Networks, which has shown exemplary performance on several competitions related to Computer Vision and Image Processing. Some of the exciting application areas of CNN include Image Classification and Segmentation, Object Detection, Video Processing, Natural Language Processing, and Speech Recognition. The powerful learning ability of deep CNN is primarily due to the use of multiple feature extraction stages that can automatically learn representations from the data. The availability of a large amount of data and improvement in the hardware technology has accelerated the research in CNNs, and recently interesting deep CNN architectures have been reported. Several inspiring ideas to bring advancements in CNNs have been explored, such as the use of different activation and loss functions, parameter optimization, regularization, and architectural innovations. However, the significant improvement in the representational capacity of the deep CNN is achieved through architectural innovations. Notably, the ideas of exploiting spatial and channel information, depth and width of architecture, and multi-path information processing have gained substantial attention. Similarly, the idea of using a block of layers as a structural unit is also gaining popularity. This survey thus focuses on the intrinsic taxonomy present in the recently reported deep CNN architectures and, consequently, classifies the recent innovations in CNN architectures into seven different categories. These seven categories are based on spatial exploitation, depth, multi-path, width, feature-map exploitation, channel boosting, and attention. Additionally, the elementary understanding of CNN components, current challenges, and applications of CNN are also provided.

Keywords Deep learning · Convolutional neural networks · Taxonomy · Representational capacity · Residual learning · Channel boosted CNN

✉ Asifullah Khan
asif@pieas.edu.pk

[1] Pattern Recognition Lab, DCIS, PIEAS, Nilore, Islamabad 45650, Pakistan

[2] Deep Learning Lab, Center for Mathematical Sciences, PIEAS, Nilore, Islamabad 45650, Pakistan

Published online: 21 April 2020

1 Introduction

Machine Learning (ML) algorithms are known to learn the underlying relationship in data and thus make decisions without requiring explicit instructions. In literature, various exciting works have been reported to understand and/or emulate the human sensory responses such as speech and vision (Hubel and Wiesel 1962, 1968; Ojala et al. 1996; Chapelle 1998; Lowe 1999; Dalal and Triggs 2004; Bay et al. 2008; Heikkilä et al. 2009). In 1989, a new class of Neural Networks (NN), called Convolutional Neural Network (CNN) (LeCun et al. 1989) was reported, which has shown enormous potential in Machine Vision (MV) related tasks.

CNNs are one of the best learning algorithms for understanding image content and have shown exemplary performance in image segmentation, classification, detection, and retrieval related tasks (Cireşan et al. 2012a, b; Liu et al. 2019). The success of CNNs has captured attention beyond academia. In industry, companies such as Google, Microsoft, AT&T, NEC, and Facebook have developed active research groups for exploring new architectures of CNN (Deng et al. 2013). At present, most of the frontrunners of image processing and computer vision (CV) competitions are employing deep CNN based models.

The attractive feature of CNN is its ability to exploit spatial or temporal correlation in data. The topology of CNN is divided into multiple learning stages composed of a combination of the convolutional layers, non-linear processing units, and subsampling layers (Jarrett et al. 2009). CNN is a feedforward multilayered hierarchical network, where each layer, using a bank of convolutional kernels, performs multiple transformations (LeCun et al. 2010). Convolution operation helps in the extraction of useful features from locally correlated data points. The output of the convolutional kernels is then assigned to the non-linear processing unit (activation function), which not only helps in learning abstractions but also embeds non-linearity in the feature space. This non-linearity generates different patterns of activations for different responses and thus facilitates in learning of semantic differences in images. The output of the non-linear activation function is usually followed by subsampling, which helps in summarizing the results and also makes the input invariant to geometrical distortions (Scherer et al. 2010; LeCun et al. 2010). CNN, with the automatic feature extraction ability, reduces the need for a separate feature extractor (Najafabadi et al. 2015). Thus, CNN without exhaustive processing can learn good internal representation from raw pixels. Notable attributes of CNN are hierarchical learning, automatic feature extraction, multi-tasking, and weight sharing (Guo et al. 2016; Liu et al. 2017; Abbas et al. 2019).

CNN first came to limelight through the work of LeCuN in 1989 for processing of grid-like topological data (images and time series data) (LeCun et al. 1989; Ian Goodfellow et al. 2017). The architectural design of CNN was inspired by Hubel and Wiesel's work and thus mostly follows the basic structure of primate's visual cortex (Hubel and Wiesel 1962, 1968). Different stages of the learning process in CNN show quite a resemblance to the primate's ventral pathway of the visual cortex (V1–V2–V3–V4–IT/VTC) (Laskar et al. 2018). The visual cortex of primates first receives input from the retinotopic area. Whereby, the lateral geniculate nucleus performs multi-scale highpass filtering and contrast normalization. After this, detection is performed by different regions of the visual cortex categorized as V1, V2, V3, and V4. In fact, V1 and V2 regions of the visual cortex are similar to convolutional and subsampling layers. In contrast, the inferior temporal region resembles the higher layers of CNN, which makes an inference about the image (Grill-Spector et al. 2018).

During training, CNN learns through backpropagation algorithm, by regulating the change in weights according to the target. Optimization of an objective function using a backpropagation algorithm is similar to the response based learning of the human brain. The multilayered, hierarchical structure of deep CNN, gives it the ability to extract low, mid, and high-level features. High-level features (more abstract features) are a combination of lower and mid-level features. The hierarchical feature extraction ability of CNN emulates the deep and layered learning process of the Neocortex in the human brain, which dynamically learns features from the raw data (Bengio 2009). The popularity of CNN is primarily due to its hierarchical feature extraction ability.

Deep architectures often have an advantage over shallow architectures when dealing with complex learning problems. The stacking of multiple linear and non-linear processing units in a layer-wise fashion provides the ability to learn complex representations at different levels of abstraction. Consequently, in recognition tasks consisting of hundreds of image categories, deep CNNs have shown substantial performance improvement over conventional vision-based models (Ojala et al. 2002; Dalal and Triggs 2004; Lowe 2004). The observation that the deep architectures can improve the representational capacity of a CNN heightened the use of CNN in image classification and segmentation tasks (Krizhevsky et al. 2012). The availability of big data and advancements in hardware are also the main reasons for the recent success of deep CNNs. Empirical studies showed that if given enough training data, deep CNNs can learn the invariant representations and may achieve human-level performance. In addition to its use as a supervised learning mechanism, the potential of deep CNNs can also be exploited to extract useful representations from a large scale of unlabeled data. Recently, it is shown that different levels of features, including both low and high-level, can be transferred to a generic recognition task by exploiting the concept of Transfer Learning (TL) (Qiang Yang et al. 2008; Qureshi et al. 2017; Qureshi and Khan 2018).

From the late 1990s up to 2000, various improvements in CNN learning methodology and architecture were performed to make CNN scalable to large, heterogeneous, complex, and multiclass problems. Innovations in CNNs include different aspects such as modification of processing units, parameter and hyper-parameter optimization strategies, design patterns and connectivity of layers, etc. CNN based applications became prevalent after the exemplary performance of AlexNet on the ImageNet dataset in 2012 (Krizhevsky et al. 2012). Significant innovations in CNN have been proposed since then and are largely attributed to the restructuring of processing units and designing of new blocks. Zeiler and Fergus (Zeiler and Fergus 2013) gave the concept of layer-wise visualization of CNN to improve the understanding of feature extraction stages, which shifted the trend towards extraction of features at low spatial resolution in deep architecture as performed in VGG (Simonyan and Zisserman 2015). Nowadays, most of the new architectures are built upon the principle of simple and homogenous topology, as introduced in VGG. Google deep learning group introduced an innovative idea of a split, transform and merge, with the corresponding block known as inception block. The inception block for the very first time gave the concept of branching within a layer, which allows abstraction of features at different spatial scales (Szegedy et al. 2015). In 2015, the concept of skip connections introduced by ResNet (He et al. 2015a) for the training of deep CNNs gained popularity. Afterward, this concept was used by most of the succeeding networks, such as Inception-ResNet, Wide ResNet, ResNeXt, etc., (Szegedy et al. 2016a; Zagoruyko and Komodakis 2016; Xie et al. 2017).

Different architectural designs such as Wide ResNet, ResNeXt, Pyramidal Net, Xception, PolyNet, and many others explore the effect of multilevel transformations on CNNs

learning capacity by introducing cardinality or increasing the width (Zagoruyko and Komodakis 2016; Zhang et al. 2017; Han et al. 2017; Xie et al. 2017). Therefore, the focus of research shifted from parameter optimization and connections readjustment towards the improved architectural design of the network. This shift resulted in many new architectural ideas such as channel boosting, spatial and feature-map wise exploitation and attention-based information processing etc., (Wang et al. 2017a; Woo et al. 2018; Khan et al. 2018a).

In the past few years, different interesting surveys are conducted on deep CNNs that elaborate on the essential components of CNN and their alternatives. The survey reported in (Gu et al. 2018) reviewed the famous architectures from 2012 to 2015 along with their basic components. Similarly, there are prominent surveys that discuss different algorithms and applications of CNN (LeCun et al. 2010; Najafabadi et al. 2015; Guo et al. 2016; Srinivas et al. 2016; Liu et al. 2017). Likewise, the survey presented in (Zhang et al. 2019) discusses the taxonomy of CNNs based on acceleration techniques. On the other hand, in this survey, we discuss the intrinsic taxonomy present in the recent and prominent CNN architectures reported from 2012 to 2020. The various CNN architectures discussed in this survey are broadly classified into seven main categories, namely; spatial exploitation, depth, multi-path, width, feature-map exploitation, channel boosting, and attention-based CNNs.

This survey also gives an insight into the basic structure of CNN as well as its historical perspective, presenting different eras of CNN that trace back from its origin to its latest developments and achievements. This survey will help the readers to develop the theoretical insight into the design principles of CNN and thus may further accelerate the architectural innovations in CNN.

The rest of the paper is organized in the following order (shown in Fig. 1): Sect. 1 develops the systematic understanding of CNN, discusses its resemblance with primate's visual cortex, as well as its contribution to MV. In this regard, Sect. 2 provides an overview of essential CNN components, and Sect. 3 discusses the architectural evolution of deep CNNs. Whereas, Sect. 4 discusses the recent innovations in CNN architectures and categorizes CNNs into seven broad classes. Sects. 5 and 6 shed light on applications of CNNs and current challenges, whereas Sect. 7 discusses future work. Finally, the last section concludes.

2 Basic CNN components

Nowadays, CNN is considered as one of the most widely used ML technique, especially in vision-related applications. CNN can learn representations from the grid-like data, and recently it has shown substantial performance improvement in various ML applications. A typical block diagram of an ML system is shown in Fig. 2. Since CNN possesses both good feature generation and discrimination ability, therefore in a typical ML system, CNN capabilities are exploited for feature generation and classification.

A typical CNN architecture generally comprises alternate layers of convolution and pooling followed by one or more fully connected layers at the end. In some cases, a fully connected layer is replaced with a global average pooling layer. In addition to different mapping functions, different regulatory units such as batch normalization and dropout are also incorporated to optimize CNN performance (Bouvrie 2006). The arrangement of CNN components plays a fundamental role in designing new

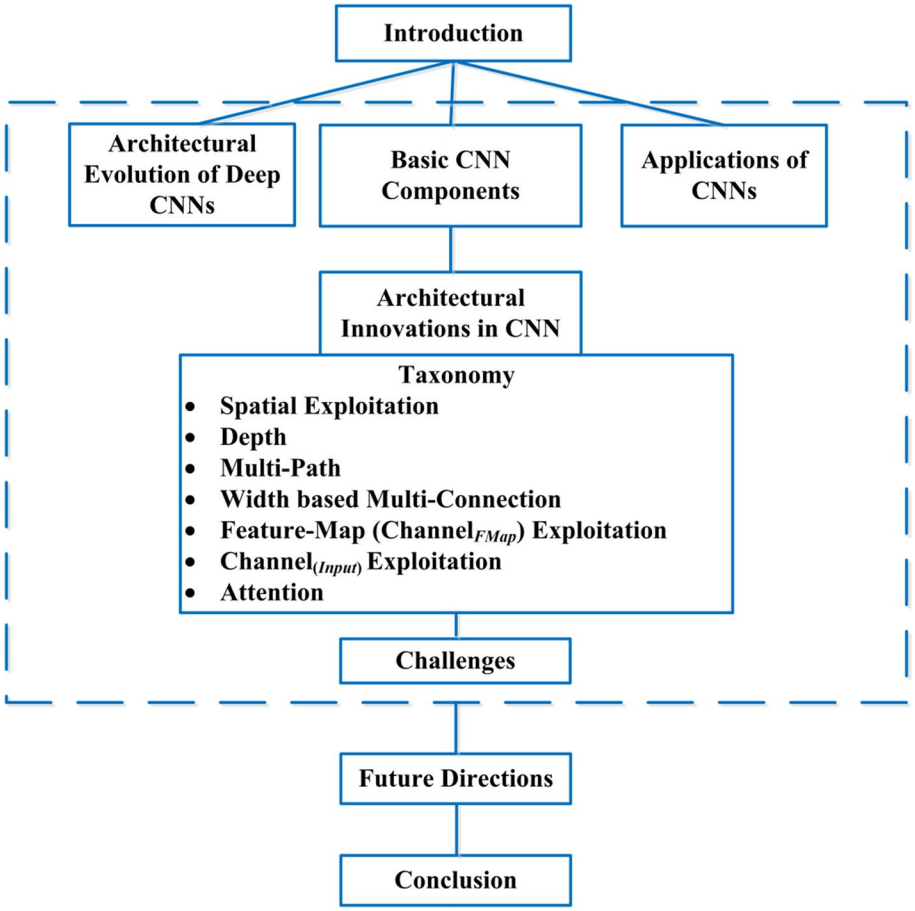

Fig. 1 Organization of the survey paper showing different sections

architectures and thus achieving enhanced performance. This section briefly discusses the role of these components in a CNN architecture.

2.1 Convolutional layer

The convolutional layer is composed of a set of convolutional kernels where each neuron acts as a kernel. However, if the kernel is symmetric, the convolution operation becomes a correlation operation (Ian Goodfellow et al. 2017). Convolutional kernel works by dividing the image into small slices, commonly known as receptive fields. The division of an image into small blocks helps in extracting feature motifs. Kernel convolves with the images using a specific set of weights by multiplying its elements with the corresponding elements of the receptive field (Bouvrie 2006). Convolution operation can be expressed as follows:

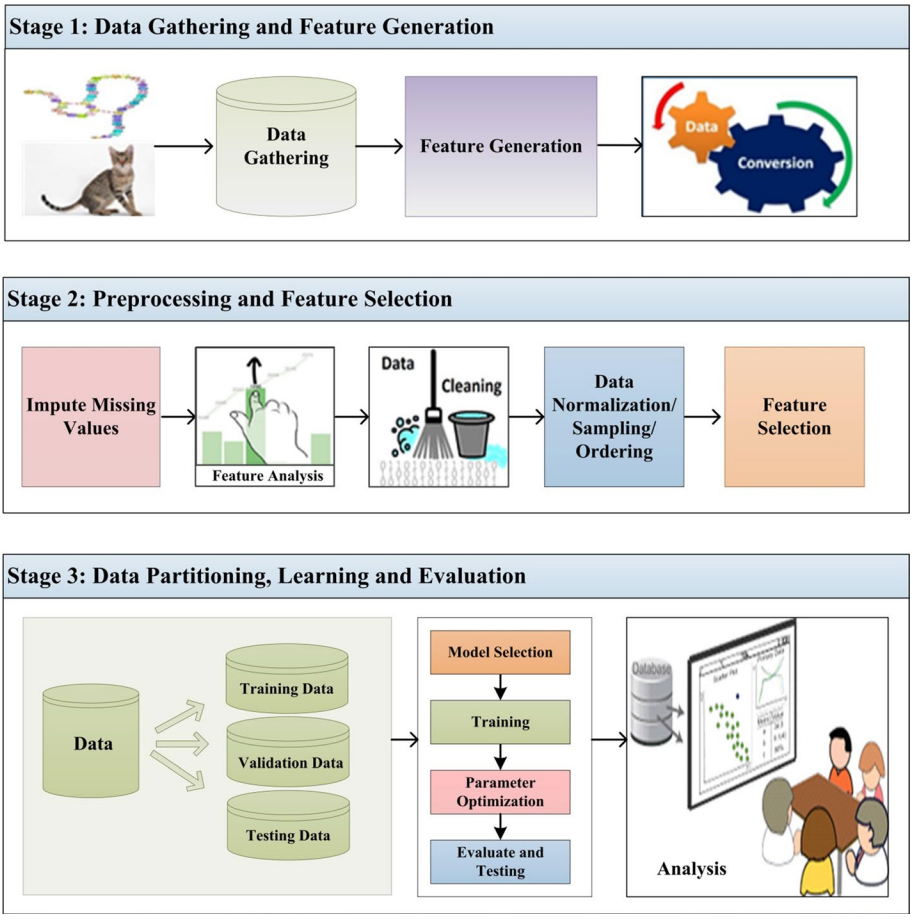

Fig. 2 Basic layout of a typical ML system having several stages

$$f_l^k(p,q) = \sum_c \sum_{x,y} i_c(x,y).e_l^k(u,v) \tag{1}$$

where $i_c(x,y)$ is an element of the input image tensor I_C, which is element wise multiplied by $e_l^k(u,v)$ index of the kth convolutional kernel k_l of the lth layer. Whereas output feature-map of the kth convolutional operation can be expressed as $\mathbf{F}_l^k = \left[f_l^k(1,1), \ldots, f_l^k(p,q), \ldots, f_l^k(P,Q)\right]$. The different mathematical symbols used are defined in Table 1.

Due to weight sharing ability of convolutional operation, different sets of features within an image can be extracted by sliding kernel with the same set of weights on the image and thus makes CNN parameter efficient as compared to the fully connected networks. Convolution operation may further be categorized into different types based on the type and size of filters, type of padding, and the direction of convolution (LeCun et al. 2015).

Table 1 Definition of mathematical symbols

Symbol	Description
X	Total x coordinates of an image
x	xth coordinate under consideration of an image
Y	Total y coordinates of an image
y	yth coordinate under consideration of an image
c	Channel index
$i_c(x, y)$	(x, y) element of cth channel of an image
L	Total number of layers
l	Layer number
K_l	Total number of kernels of lth layer
k_l	Kernel number of lth layer
U	Total number of rows of kth kernel
u	uth row under consideration
V	Total number of columns of kth kernel
v	vth column under consideration
$e_l^k(u, v)$	(u, v) element of kth kernel of lth layer
$\mathbf{F}_l^k$	Input feature matrix for lth layer and kth neuron
P	Total number of rows of feature matrix
p	pth row under consideration
Q	Total number of columns of feature matrix
q	qth column under consideration
$f_l^k(p, q)$	(p, q) element of feature matrix
$g_c(.)$	Convolution operation
$g_p(.)$	Pooling operation
$g_a(.)$	Activation function
$g_k(.)$	Concatenation operation
g_{t_g}	Transformation gate
g_{c_g}	Carry gate
$g_{sq}(.)$	Squeeze operation
$g_{ex}(.)$	Excitation operation
$\mathbf{Y}_{l+1}^K$	Weight vector showing feature-maps importance learned using SE operation
g_t	Transformation function for two layer NN implemented by SE block
g_{s_g}	Sigmoid gate implemented by SE block
g_{sm}	Soft mask
g_{tm}	Trunk mask
$\mathbf{I}_B$	Channel boosted input tensor

2.2 Pooling layer

Feature motifs, which result as an output of convolution operation, can occur at different locations in the image. Once features are extracted, its exact location becomes less important as long as its approximate position relative to others is preserved. Pooling or down-sampling is an interesting local operation. It sums up similar information in the

neighborhood of the receptive field and outputs the dominant response within this local region (Lee et al. 2016).

$$\mathbf{Z}_l^k = g_p(\mathbf{F}_l^k) \tag{2}$$

Equation (2) shows the pooling operation in which $\mathbf{Z}_l^k$ represents the pooled feature-map of lth layer for kth input feature-map $\mathbf{F}_l^k$, whereas $g_p(.)$ defines the type of pooling operation.

The use of pooling operation helps to extract a combination of features, which are invariant to translational shifts and small distortions (Huang et al. 2007; Scherer et al. 2010). Reduction in the size of feature-map to invariant feature set not only regulates the complexity of the network but also helps in increasing the generalization by reducing overfitting. Different types of pooling formulations such as max, average, L2, overlapping, spatial pyramid pooling, etc. are used in CNN (Boureau 2009; Wang et al. 2012; He et al. 2015b).

2.3 Activation function

Activation function serves as a decision function and helps in learning of intricate patterns. The selection of an appropriate activation function can accelerate the learning process. The activation function for a convolved feature-map is defined in Eq. (3).

$$\mathbf{T}_l^k = g_a(\mathbf{F}_l^k) \tag{3}$$

In the above equation, $\mathbf{F}_l^k$ is an output of a convolution, which is assigned to activation function $g_a(.)$ that adds non-linearity and returns a transformed output $\mathbf{T}_l^k$ for lth layer. In literature, different activation functions such as sigmoid, tanh, maxout, SWISH, ReLU, and variants of ReLU, such as leaky ReLU, ELU, and PReLU are used to inculcate non-linear combination of features (LeCun 2007; Wang et al. 2012; Xu et al. 2015a; Ramachandran et al. 2017; Gu et al. 2018). However, ReLU and its variants are preferred as they help in overcoming the vanishing gradient problem (Hochreiter 1998; Nwankpa et al. 2018). One of the recently proposed activation function is MISH, which has shown better performance than ReLU in most of the recently proposed deep networks on benchmark datasets (Misra 2019).

2.4 Batch normalization

Batch normalization is used to address the issues related to the internal covariance shift within feature-maps. The internal covariance shift is a change in the distribution of hidden units' values, which slows down the convergence (by forcing learning rate to small value) and requires careful initialization of parameters. Batch normalization for a transformed feature-map $\mathbf{F}_l^k$ is shown in Eq. (4).

$$\mathbf{N}_l^k = \frac{\mathbf{F}_l^k - \mu_B}{\sqrt{\sigma_B^2 + \varepsilon}} \tag{4}$$

In Eq. (4), $\mathbf{N}_l^k$ represents normalized feature-map, $\mathbf{F}_l^k$ is the input feature-map, μ_B and σ_B^2 depict mean and variance of a feature-map for a mini batch respectively. In order to avoid division by zero, ε is added for numerical stability. Batch normalization unifies the

distribution of feature-map values by setting them to zero mean and unit variance (Ioffe and Szegedy 2015). Furthermore, it smoothens the flow of gradient and acts as a regulating factor, which thus helps in improving the generalization of the network.

2.5 Dropout

Dropout introduces regularization within the network, which ultimately improves generalization by randomly skipping some units or connections with a certain probability. In NNs, multiple connections that learn a non-linear relation are sometimes co-adapted, which causes overfitting (Hinton et al. 2012b). This random dropping of some connections or units produces several thinned network architectures, and finally, one representative network is selected with small weights. This selected architecture is then considered as an approximation of all of the proposed networks (Srivastava et al. 2014).

2.6 Fully connected layer

Fully connected layer is mostly used at the end of the network for classification. Unlike pooling and convolution, it is a global operation. It takes input from feature extraction stages and globally analyses the output of all the preceding layers (Lin et al. 2013). Consequently, it makes a non-linear combination of selected features, which are used for the classification of data (Rawat and Wang 2016).

3 Architectural evolution of deep CNNs

Nowadays, CNNs are considered as the most widely used algorithms among biologically inspired Artificial Intelligence (AI) techniques. CNN history begins with the neurobiological experiments conducted by Hubel and Wiesel (1959, 1962) (Hubel and Wiesel 1959, 1962). Their work provided a platform for many cognitive models, and CNN replaced almost all of these. Over the decades, different efforts have been carried out to improve the performance of CNNs. The evolutionary history of deep CNN architectures is pictorially represented in Fig. 3. Improvements in CNN architectures can be categorized into five different eras that are discussed below.

3.1 Origin of CNN: late 1980s–1999

CNNs have been applied to visual tasks since the late 1980s. In 1989, LeCuN et al. proposed the first multilayered CNN named ConvNet, whose origin rooted in Fukushima's Neocognitron (Fukushima and Miyake 1982; Fukushima 1988). LeCuN proposed a supervised training of ConvNet using the backpropagation algorithm, in comparison to the unsupervised reinforcement learning scheme used by its predecessor Neocognitron (Linnainmaa 1970; LeCun et al. 1989). LeCuN's work thus made a foundation for the modern 2D CNNs. This ConvNet showed successful results for handwritten digit and zip code recognition related problems (Zhang and LeCun 2015). In 1998, LeCuN proposed an improved version of ConvNet, which was famously known as LeNet-5, and it started the use of CNN in classifying characters in a document recognition related applications (LeCun et al. 1995,

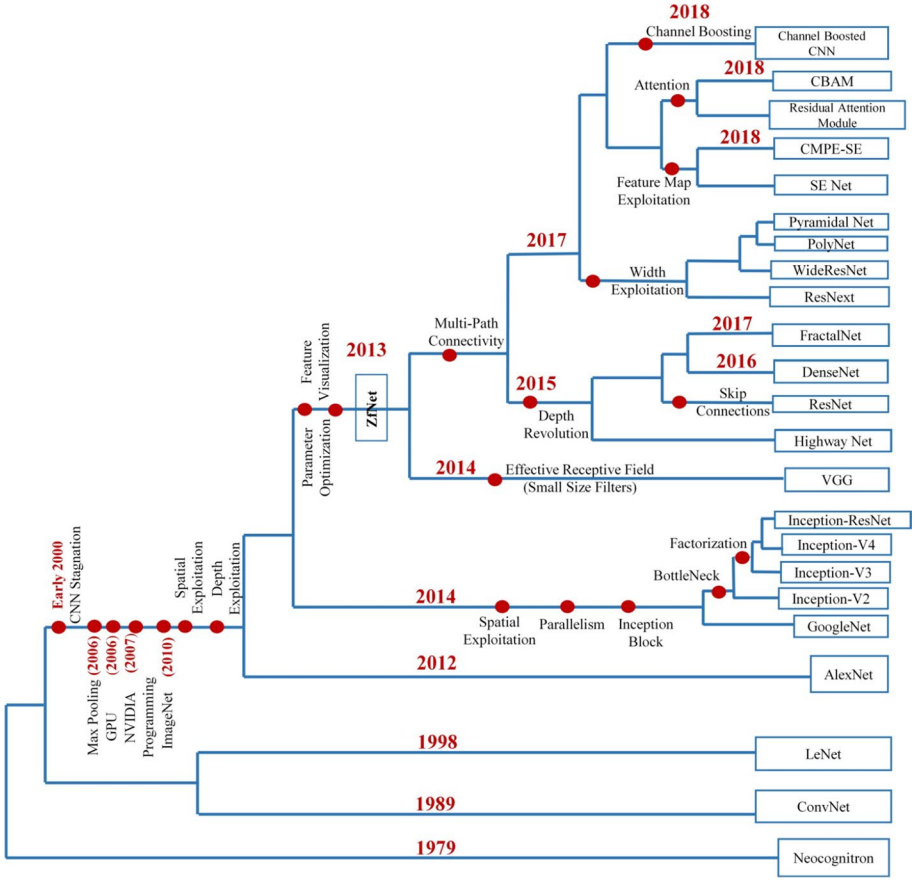

Fig. 3 Evolutionary history of deep CNNs showing architectural innovations from ConvNet till to date architectures

1998). Due to the good performance of CNN in optical character and fingerprint recognition, its commercial use in ATM and Banks started in 1993 and 1996, respectively. In this era, LeNet-5 achieved many successful milestones for optical character recognition tasks, but it didn't perform well on other image recognition problems.

3.2 Stagnation of CNN: early 2000

In the late 1990s and early 2000, researchers had little insight into the internal working of CNN, and it was considered as a black box. Complicated architecture design and heavy processing made it hard to train CNN. It was widely presumed in early 2000 that the back-propagation algorithm used for training of CNN was not effective in converging to the global minima of the error surface. Thus, CNN was considered as a less effective feature extractor compared to handcrafted features (Schmidhuber 2007). Moreover, no comprehensive dataset of diverse categories of images was available at that time. Therefore, because of the insignificant improvement in CNN performance at the cost of high computational time, little attention was given to explore its role in different applications such as object

detection, video surveillance, etc. At that time, other statistical methods and in particular, SVM became more popular than CNN due to their relatively high performance (Joachims 1998; Decoste and Schölkopf 2002; Liu et al. 2003).

Meanwhile, a few research groups kept on working with CNNs and tried to optimize its performance. In 2003, Simard et al. improved CNN architecture and showed good results compared to SVM on a hand digit benchmark dataset; MNIST (LeCun et al. 1998; Liu et al. 2003; Simard et al. 2003; Chellapilla et al. 2006; Deng 2012). This improvement in performance expedited the research in CNNs by extending their application's beyond optical character recognition to other script's character recognition, deployment in image sensors for face detection in video conferencing, and regulation of street crimes, etc. (Abdulkader 2006; Chellapilla et al. 2006; Cireşan et al. 2010). Likewise, CNN based systems were industrialized in markets for customers' tracking (Garcia and Delakis 2004; Frome et al. 2009; LeCun et al. 2010). Moreover, CNN's potential in other applications such as medical image segmentation, anomaly detection, and robot vision was also explored (Fasel 2002; Matsugu et al. 2002; Chen et al. 2006).

3.3 Revival of CNN: 2006–2011

Deep CNNs generally have complex architecture and time-intensive training phase that sometimes may span over weeks. In early 2000, there were a few parallel processing techniques and limited hardware resources for the training of deep Networks. Training of a deep CNNs with a typical activation function such as sigmoid may suffer from exponential decay and explosion of a gradient. Since 2006, significant efforts have been made to tackle the CNN optimization problem. In this regard, several interesting initialization and training strategies were reported to overcome the difficulties encountered in the training of deep CNNs and the learning of invariant features. Hinton reported the concept of greedy layer-wise pre-training in 2006, which revived the research in deep learning (Hinton et al. 2006; Khan et al. 2018b). Experimental studies showed that both supervised and unsupervised pre-training could initialize a network in a better way than random initialization. Bengio and other researchers proposed that the sigmoid activation function is not suitable for the training of deep architectures with random initialization of weights. This observation started the use of activation functions other than sigmoid such as ReLU, tanh etc., (Glorot and Bengio 2010). The revival of deep learning was one of the factors, which brought deep CNNs into limelight (Bengio et al. 2007, 2013).

Ranzato et al. (2007) used max-pooling instead of subsampling, which showed good results by learning invariant features (Ranzato et al. 2007; Giusti et al. 2013). In late 2006, researchers started using graphics processing units (GPUs) to accelerate the training of deep NN and CNN architectures (Oh and Jung 2004; Strigl et al. 2010; Cireşan et al. 2011; Nguyen et al. 2019). In 2007, NVIDIA launched the CUDA programming platform, which allows exploitation of parallel processing capabilities of GPU with a greater degree (Nickolls et al. 2008; Lindholm et al. 2008). In essence, the use of GPUs for NN and CNN training and other hardware improvements were the main factors, which revived the research in CNN (Oh and Jung 2004; Cireşan et al. 2018). In 2010, Fei–Fei Li's group at Stanford, established a large database of images known as ImageNet, containing millions of annotated images belonging to a large number of classes (Russakovsky et al. 2015). This database was coupled with the annual ImageNet Large Scale Visual Recognition Challenge (ILSVRC), where the performances of various models have been evaluated and scored (Berg et al. 2010). Similarly, in the same year, Stanford released PASCAL 2010 VOC

dataset for object detection. ILSVRC and Neural Information Processing Systems Conference (NIPS) are the two platforms that play a dominant role in strengthening research and increasing the use of CNN and thus making it popular.

3.4 Rise of CNN: 2012–2014

The availability of extensive training data and hardware advancements are the factors that contributed to the advancement in CNN research. But the main driving forces that have accelerated the research and give rise to the use of CNNs in image classification and recognition tasks are parameter optimization strategies and new architectural ideas (Gu et al. 2018; Sinha et al. 2018; Zhang et al. 2019). The main breakthrough in CNN performance was brought by AlexNet, which showed exemplary performance in 2012-ILSVRC (reduced error rate from 25.8 to 16.4) as compared to conventional CV techniques (Krizhevsky et al. 2012).

In this era, several attempts were made to improve the performance of CNN; depth and parameter optimization strategies were explored with a significant reduction in computational cost. Similarly, different architectural designs were proposed, whereby each new architecture tried to overcome the shortcomings of previously proposed architectures in combination with new structural reformulations. With the trend of designing very deep CNNs, it generally becomes difficult to independently determine filter dimensions, stride, padding, and other hyper-parameters for each layer. This problem is resolved by designing convolutional layers with a fixed topology that can be repeated multiple times. This shifted the trend from custom layer design towards modular and uniform layer design. The concept of modularity in CNNs made it easy to tailor them for different tasks effortlessly (Simonyan and Zisserman 2015; Amer and Maul 2019). In this connection, a different idea of branching and block within a layer was introduced by the Google group (Szegedy et al. 2015). It should be noted that in this era, two different types of architectures, deep and narrow, as well as deep and wide, were in use.

3.5 Rapid increase in architectural innovations and applications of CNN: 2015-present

The research in CNN is still going on and has a significant potential for improvement. It is generally observed that the significant improvements in CNN performance occurred from 2015 to 2019. The representational capacity of a CNN usually depends on its depth, and in a sense, an enriched feature set ranging from simple to complex abstractions can help in learning complex problems. However, the main challenge faced by deep architectures is that of the diminishing gradient. Initially, researchers tried to subside this problem by connecting intermediate layers to auxiliary learners (Szegedy et al. 2015). In 2015, the emerging area of research was mainly the development of new connections to improve the convergence rate of deep CNN architectures. In this regard, different ideas such as information gating mechanism across multiple layers, skip connections, and cross-layer channel connectivity was introduced (Srivastava et al. 2015a; He et al. 2015a; Huang et al. 2017). Different experimental studies showed that state-of-the-art deep architectures such as VGG, ResNet, ResNext, etc. also showed good results for challenging recognition and localization problems like semantic and instance-based object segmentation, scene parsing, scene location, etc. Most of the famous object detection and segmentation architectures such

as Single Shot Multibox Detector (SSD), Region-based CNN (R-CNN), Faster R-CNN, Mask R-CNN and Fully Convolutional Neural Network (FCN) are built on the lines of ResNet, VGG, Inception, etc. Similarly, many interesting detection algorithms such as Feature Pyramid Networks, Cascade R-CNN, Libra R-CNN, etc., modified the architectures as mentioned earlier to improve the performance (Lin et al. 2017; Cai and Vasconcelos 2019; Pang et al. 2020). Applications of deep CNN were also extended to image captioning by combining these networks with recurrent neural network (RNN) and thus showed state-of-the-art results on MS COCO-2015 image captioning challenge (Girshick 2015; Long et al. 2015; Ren et al. 2015; He et al. 2017; Vinyals et al. 2017).

Similarly, in 2016, it was observed that the stacking of multiple transformations not only depth-wise but also in parallel fashion showed good learning for complex problems (Zagoruyko and Komodakis 2016; Han et al. 2017). Different researchers used a hybrid of the already proposed architectures to improve deep CNN performance (Huang et al. 2016a; Szegedy et al. 2016a; Targ et al. 2016; Yamada et al. 2016; Kuen et al. 2017; Lv et al. 2019). In 2017, the focus of researchers was mainly on designing of generic blocks that can be inserted at any learning stage in CNN architecture to improve the network representation (Hu et al. 2018a). Designing of new blocks is one of the growing areas of research in CNN, where generic blocks are used to assign attention to spatial and feature-map (channel) information (Wang et al. 2017a; Roy et al. 2018; Woo et al. 2018). In 2018, a new idea of channel boosting was introduced by Khan et al. (2018a) to boost the performance of a CNN by learning distinct features as well as exploiting the already learned features through the concept of TL.

However, two main concerns observed with deep and wide architectures are the high computational cost and memory requirement. As a result, it is very challenging to deploy state-of-the-art wide and deep CNN models in resource-constrained environments. Conventional convolution operation requires a huge number of multiplications, which increases the inference time and restricts the applicability of CNN to low memory and time constraint applications (Shakeel et al. 2019). Many real-world applications, such as autonomous vehicles, robotics, healthcare, and mobile applications, perform the tasks that need to be carried on computationally limited platforms in a timely manner. Therefore, different modifications in CNN are performed to make them appropriate for resource-constrained environments. Prominent modifications are knowledge distillation, training of small networks, or squeezing of pre-trained networks (such as pruning, quantization, hashing, Huffman coding, etc.) (Chen et al. 2015; Han et al. 2016; Wu et al. 2016; Frosst and Hinton 2018). GoogleNet exploited the idea of small networks, which replaces the conventional convolution with point-wise group convolution operation to make it computationally efficient. Similarly, ShuffleNet used point-wise group convolution but with a new idea of channel shuffle that significantly reduces the number of operations without affecting the accuracy. In the same way, ANTNet proposed a novel architectural block known as ANT-Block, which at low computational cost, achieved good performance on benchmark datasets (Howard et al. 2017; Zhang et al. 2018a; Xiong et al. 2019).

From 2012 up till now, many improvements have been reported in CNN architectures. As regards the architectural advancement of CNNs, recently, the focus of research has been on designing of new blocks that can boost network representation by exploiting feature-maps or manipulating input representation by adding artificial channels. Moreover, along with this, the trend is towards the design of lightweight architectures without compromising the performance to make CNN applicable for resource constraint hardware.

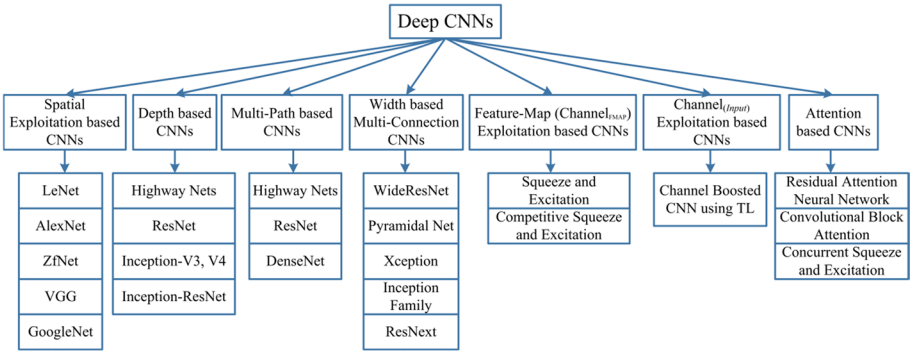

Fig. 4 Taxonomy of deep CNN architectures showing seven different categories

4 Architectural innovations in CNN

Different improvements in CNN architecture have been made from 1989 to date. These improvements can be categorized as parameter optimization, regularization, structural reformulation, etc. However, it is observed that the main thrust in CNN performance improvement came from the restructuring of processing units and the designing of new blocks. Most of the innovations in CNN architectures have been made in relation to depth and spatial exploitation. Depending upon the type of architectural modifications, CNNs can be broadly categorized into seven different classes, namely; spatial exploitation, depth, multi-path, width, feature-map exploitation, channel boosting, and attention-based CNNs. The taxonomy of CNN architectures is pictorially represented in Fig. 4. Architectural details of the state-of-the-art CNN models, their parameters, and performance on benchmark datasets are summarized in Table 2. On the other hand, different online resources on deep CNN architectures, vision-related dataset, and their implementation platforms are mentioned in Table 3. In addition to this, the strengths and weaknesses of various architectures based on their category are presented in Tables 5, 6, 7, 8, 9, 10, 11.

4.1 Spatial exploitation based CNNs

CNNs have a large number of parameters and hyper-parameters, such as weights, biases, number of layers, and processing units (neurons), filter size, stride, activation function, learning rate, etc. (Kafi et al. 2015; Shin et al. 2016). As convolutional operation considers the neighborhood (locality) of input pixels, therefore different levels of correlation can be explored by using different filter sizes. Different sizes of filters encapsulate different levels of granularity; usually, small size filters extract fine-grained and large size extract coarse-grained information. Consequently, in early 2000, researchers exploited spatial filters to improve performance and explored the relation of a spatial filter with the learning of the network. Different studies conducted in this era suggested that by the adjustment of filters, CNN can perform well both on coarse and fine-grained details.

Table 2 Performance comparison of the recent architectures of different categories. Top 5 error rate is reported for all architectures

Architecture name	Year	Main contribution	Parameters	Error rate	Depth	Category	References
LeNet	1998	First popular CNN architecture	0.060 M	[dist]MNIST: 0.8 MNIST: 0.95	5	Spatial exploitation	LeCun et al. (1995)
AlexNet	2012	Deeper and wider than the LeNet Uses Relu, dropout and overlap Pooling GPUs NVIDIA GTX 580	60 M	ImageNet: 16.4	8	Spatial Exploitation	Krizhevsky et al. (2012)
ZfNet	2014	Visualization of intermediate layers	60 M	ImageNet: 11.7	8	Spatial exploitation	Zeiler and Fergus (2013)
VGG	2014	Homogenous topology Uses small size kernels	138 M	ImageNet: 7.3	19	Spatial exploitation	Simonyan and Zisserman (2015)
GoogLeNet	2015	Introduced block concept Split transform and merge idea	4 M	ImageNet: 6.7	22	Spatial exploitation	Szegedy et al. (2015)
Inception-V3	2015	Handles the problem of a representational bottleneck Replace large size filters with small filters	23.6 M	ImageNet: 3.5 Multi-crop: 3.58 Single-Crop: 5.6	159	Depth + width	Szegedy et al. (2016b)
Highway networks	2015	Introduced an idea of multi-path	2.3 M	CIFAR-10: 7.76	19	Depth + multi-path	Srivastava et al. (2015a)
Inception-V4	2016	Split transform and merge idea Uses asymmetric filters	35 M	ImageNet: 4.01	70	Depth +width	Szegedy et al. (2016a)
Inception-ResNet	2016	Uses split transform merge idea and residual links	55.8 M	ImageNet: 3.52	572	Depth + width + multi-path	Szegedy et al. (2016a)
ResNet	2016	Residual learning Identity mapping based skip connections	25.6 M 1.7 M	ImageNet: 3.6 CIFAR-10: 6.43	152 110	Depth + multi-path	He et al. (2015a)
DelugeNet	2016	Allows cross layer information flow in deep networks	20.2 M	CIFAR-10: 3.76 CIFAR-100: 19.02	146	Multi-path	Kuen et al. (2018)

Table 2 (continued)

Architecture name	Year	Main contribution	Parameters	Error rate	Depth	Category	References
FractalNet	2016	Different path lengths are interacting 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	20 40	Multi-path	Larsson et al. (2016)
WideResNet	2016	Width is increased and depth is decreased	36.5 M	CIFAR-10: 3.89 CIFAR-100: 18.85	28 –	Width	Zagoruyko and Komodakis (2016)
Xception	2017	Depth wise convolution followed by point wise convolution	22.8 M	ImageNet: 0.055	126	Width	Chollet (2017)
Residual attention neural network	2017	Introduced an attention mechanism	8.6 M	CIFAR-10: 3.90 CIFAR-100: 20.4 ImageNet: 4.8	452	Attention	Wang et al. (2017a)
ResNeXt	2017	Cardinality Homogeneous topology Grouped convolution	68.1 M	CIFAR-10: 3.58 CIFAR-100: 17.31 ImageNet: 4.4	29 - 101	Width	Xie et al. (2017)
Squeeze and excitation networks	2017	Models interdependencies between feature-maps	27.5 M	ImageNet: 2.3	152	Feature-map exploitation	Hu et al. (2018a)
DenseNet	2017	Cross-layer 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	190 190 250 250	Multi-path	Huang et al. (2017)
PolyNet	2017	Experimented structural diversity Introduced poly inception module Generalizes residual unit using polynomial compositions	92 M	ImageNet: Single:4.25 Multi:3.45	– –	Width	Zhang et al. (2017)

Table 2 (continued)

Architecture name	Year	Main contribution	Parameters	Error rate	Depth	Category	References
PyramidalNet	2017	Increases width gradually per unit	116.4 M 27.0 M 27.0 M	ImageNet: 4.7 CIFAR-10: 3.48 CIFAR-100: 17.01	200 164 164	Width	Han et al. (2017)
Convolutional block attention Module (ResNeXt101 (32 × 4d) + CBAM)	2018	Exploits both spatial and feature-map information	48.96 M	ImageNet: 5.59	101	Attention	Woo et al. (2018)
Concurrent spatial and channel excitation mechanism	2018	Spatial attention Feature-map attention Concurrent placement of spatial and channel attention	–	MALC: 0.12 Visceral: 0.09	–	Attention	Roy et al. (2018)
Channel boosted CNN	2018	Boosting of original channels with additional information rich generated artificial channels	–	–	–	Channel boosting	Khan et al. (2018a)
Competitive squeeze and excitation network CMPE-SE-WRN-28	2018	Residual and identity mappings both are used for rescaling the feature-map	36.92 M 36.90 M	CIFAR-10: 3.58 CIFAR-100: 18.47	152 152	Feature-map exploitation	Hu et al. (2018b)

Table 3 Different online available resources for deep CNNs

Category	Description	Source
Cloud based platforms	Online free access to GPU and other deep learning accelerators	Google Colab: https://colab.research.google.com/notebooks/welcome.ipynb
		Colac: https://cocalc.com/
	Commercial platforms offered by world leading companies	FloydHub: https://www.floydhub.com/
		Amazon SageMaker: https://aws.amazon.com/deep-learning/
		Microsoft Azure ML Services: https://azure.microsoft.com/en-gb/services/machine-learning/
		Google Cloud: https://cloud.google.com/deep-learning-vm/
		IBM Watson Studio: https://www.ibm.com/cloud/deep-learning
Deep learning libraries	Deep learning libraries that provide built-in classes of NNs, fast numerical computation and automated estimation of gradients both for CPU and GPU	Pytorch: https://pytorch.org/
		Tensorflow: https://www.tensorflow.org/
		MatConvNet: http://www.vlfeat.org/matconvnet/
		Keras: https://keras.io/
		Theano: http://deeplearning.net/software/theano/
		Caffe: https://caffe.berkeleyvision.org/
		Julia: https://julialang.org/

Table 3 (continued)

Category	Description	Source
Lecture series	Online available and freely accessible deep learning courses	Stanford Lecture Series: http://cs231n.stanford.edu/ Udacity: https://www.udacity.com/course/deep-learning-nanodegree–nd101 https://www.udacity.com/course/deep-learning-pytorch–ud188 Udemy: https://www.udemy.com/course/deep-learning-learn-cnns/ https://www.udemy.com/course/modern-deep-convolutional-neural-networks/ https://www.udemy.com/course/advanced-computer-vision/ https://www.udemy.com/course/deep-learning-convolutional-neural-networks-theano-tensorflow/ https://www.udemy.com/course/deep-learning-pytorch/ Coursera: https://www.coursera.org/learn/convolutional-neural-networks https://www.coursera.org/specializations/deep-learning

Table 3 (continued)

Category	Description	Source
Vision datasets	Online freely accessible datasets of different categories of annotated images	ImageNet: http://image-net.org/
		COCO: http://cocodataset.org/#home
		Visual Genome: http://visualgenome.org/
		Open images: https://ai.googleblog.com/2016/09/introducing-open-images-dataset.html
		Places: http://places.csail.mit.edu/index.html
		Youtube-8 M: https://research.google.com/youtube8m/index.html
		CelebA: http://mmlab.ie.cuhk.edu.hk/projects/CelebA.html
		CIFAR10: https://www.cs.toronto.edu/~kriz/cifar.html
		Indoor Scene Recognition: http://web.mit.edu/torralba/www/indoor.html
		Computer Vision Datasets: https://computervisiononline.com/datasets
		Fashion MNIST: https://research.zalando.com/welcome/mission/research-projects/fashion-mnist/
Deep learning accelerators	Energy and computation efficient deep learning accelerators	NVIDIA: http://nvdla.org/
		FPGA: https://www.intel.com/content/www/us/en/artificial-intelligence/programmable/overview.html
		Eyeriss: http://eyeriss.mit.edu/
		Google's TPU: https://cloud.google.com/tpu/

4.1.1 LeNet

LeNet was proposed by LeCuN in 1998 (LeCun et al. 1995). It is famous due to its historical importance as it was the first CNN, which showed state-of-the-art performance on hand digit recognition tasks. It has the ability to classify digits without being affected by small distortions, rotation, and variation of position and scale. LeNet is a feed-forward NN that constitutes of five alternating layers of convolutional and pooling, followed by two fully connected layers. In early 2000, GPU was not commonly used to speed up training, and even CPUs were slow (Potluri et al. 2011). The main limitation of traditional multilayered fully connected NN was that it considers each pixel as separate input and applies a transformation on it, which was a substantial computational burden, specifically at that time (Gardner and Dorling 1998). LeNet exploited the underlying basis of the image that the neighboring pixels are correlated to each other and feature motifs are distributed across the entire image. Therefore, convolution with learnable parameters is an effective way to extract similar features at multiple locations with few parameters. Learning with sharable parameters changed the conventional view of training where each pixel was considered as a separate input feature from its neighborhood and ignored the correlation among them. LeNet was the first CNN architecture, which not only reduced the number of parameters but was able to learn features from raw pixels automatically.

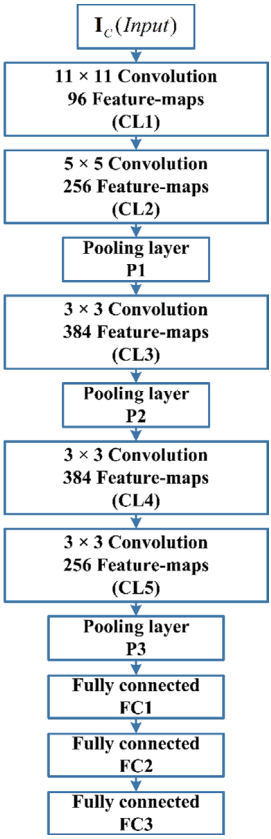

Fig. 5 Basic layout of AlexNet architecture showing its five convolution and three fully connected layers

4.1.2 AlexNet

LeNet (LeCun et al. 1995) though, begin the history of deep CNNs, but at that time, CNN was limited to hand digit recognition tasks and didn't perform well to all classes of images. AlexNet (Krizhevsky et al. 2012) is considered as the first deep CNN architecture, which showed groundbreaking results for image classification and recognition tasks. AlexNet was proposed by Krizhevsky et al. (2012) which enhanced the learning capacity of the CNN by making it deeper and by applying several parameter optimizations strategies. The basic architectural design of AlexNet is shown in Fig. 5. In early 2000, hardware limitations curtailed the learning capacity of deep CNN architectures by restricting them to small size. In order to get the benefit of the representational capacity of deep CNNs, Alexnet was trained in parallel on two NVIDIA GTX 580 GPUs to overcome shortcomings of the hardware.

In AlexNet, depth was extended from 5 (LeNet) to 8 layers to make CNN applicable for diverse categories of images. Despite the fact that generally, depth improves generalization for different resolutions of images but, the main drawback associated with an increase in depth is overfitting. To address this challenge, Krizhevesky et al. (2012) exploited the idea of Hinton (Dahl et al. 2013; Srivastava et al. 2014), whereby their algorithm randomly skips some transformational units during training to enforce the model to learn more robust features. In addition to this, ReLU was employed as a non-saturating activation function to improve the convergence rate by alleviating the problem of vanishing gradient to some extent (Hochreiter 1998; Nair and Hinton 2010). Overlapping subsampling and local response normalization were also applied to improve the generalization by reducing overfitting. Other adjustments made were the use of large size filters (11×11 and 5×5) at the initial layers, compared to previously proposed networks. Due to the efficient learning approach of AlexNet, it has significant importance in the new generation of CNNs and has started a new era of research in the architectural advancements of CNNs.

4.1.3 ZfNet

The learning mechanism of CNN, before 2013, was based mainly on hit-and-trial, without knowing the exact reason behind the improvement. This lack of understanding limited the performance of deep CNNs on complex images. In 2013, Zeiler and Fergus proposed an interesting multilayer Deconvolutional NN (DeconvNet), which got famous as ZfNet (Zeiler and Fergus 2013). ZfNet was developed to visualize network performance quantitatively. The idea of the visualization of network activity was to monitor CNN performance by interpreting neuron's activation. In one of the previous studies, Erhan et al. (2009) exploited the same idea and optimized the performance of Deep Belief Networks (DBNs) by visualizing the hidden layers' feature (Erhan et al. 2009). Similarly, Le et al. (2011) evaluated the learning of deep unsupervised autoencoder (AE) by visualizing the image classes generated by the neurons of last layer. DeconvNet works in the same manner as the forward pass CNN but reverses the order of convolutional and pooling operation. This reverse mapping projects the output of the convolutional layer back to visually perceptible image patterns, consequently gives the neuron-level interpretation of the internal feature representation learned at each layer (Simonyan et al. 2013; Grün et al. 2016).

The idea of feature visualization proposed by ZfNet was experimentally validated on AlexNet using DeconvNet, which showed that only a few neurons were active. In contrast, other neurons were dead (inactive) in the first and second layers of the network. Moreover,

it showed that the features extracted by the second layer exhibited aliasing artifacts. Based on these findings, Zeiler and Fergus adjusted CNN topology and performed parameter optimization. Zeiler and Fergus maximized the learning of CNN by reducing both the filter size and stride to retain the maximum number of features in the first two convolutional layers. This readjustment in CNN topology resulted in performance improvement, which suggested that features visualization can be used for the identification of design shortcomings and for timely adjustment of parameters.

4.1.4 VGG

The successful use of CNNs in image recognition tasks has accelerated the research in architectural design. In this regard, Simonyan et al. proposed a simple and effective design principle for CNN architectures. Their architecture, named as VGG, was modular in layers pattern (Simonyan and Zisserman 2015). VGG was made 19 layers deep compared to AlexNet and ZfNet to simulate the relation of depth with the representational capacity of the network (Krizhevsky et al. 2012; Zeiler and Fergus 2013). ZfNet, which was a frontline network of 2013-ILSVRC competition, suggested that small size filters can improve the performance of the CNNs. Based on these findings, VGG replaced the 11×11 and 5×5 filters with a stack of 3×3 filters layer and experimentally demonstrated that concurrent placement of small size (3×3) filters could induce the effect of the large size filter (5×5 and 7×7). The use of the small size filters provides an additional benefit of low computational complexity by reducing the number of parameters. These findings set a new trend in research to work with smaller size filters in CNN. VGG regulates the complexity of a network by placing 1×1 convolutions in between the convolutional layers, which, besides, learn a linear combination of the resultant feature-maps. For the tuning of the network, max-pooling is placed after the convolutional layer, while padding was performed to maintain the spatial resolution (Huang et al. 2007). VGG showed good results both for image classification and localization problems. VGG was at 2nd place in the 2014-ILSVRC competition but, got fame due to its simplicity, homogenous topology, and increased depth. The main limitation associated with VGG was the use of 138 million parameters, which make it computationally expensive and difficult to deploy it on low resource systems.

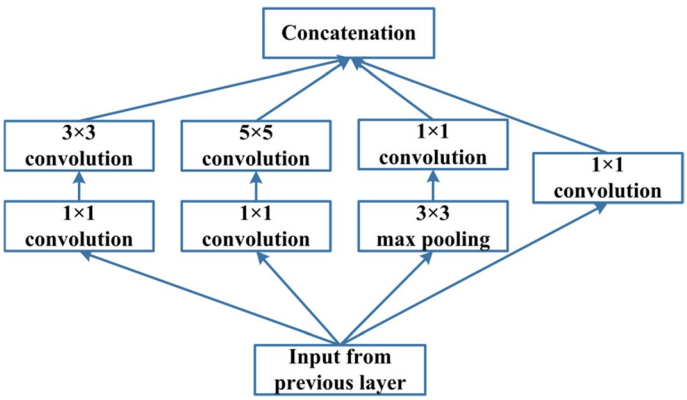

Fig. 6 Basic architecture of the inception block showing the split, transform, and merge concept

4.1.5 GoogleNet

GoogleNet was the winner of the 2014-ILSVRC competition and is also known as Inception-V1. The main objective of the GoogleNet architecture was to achieve high accuracy with a reduced computational cost (Szegedy et al. 2015). It introduced the new concept of inception block in CNN, whereby it incorporates multi-scale convolutional transformations using split, transform and merge idea. The architecture of the inception block is shown in Fig. 6. In GoogleNet, conventional convolutional layers are replaced in small blocks similar to the idea of substituting each layer with micro NN as proposed in Network in Network (NIN) architecture (Lin et al. 2013). This block encapsulates filters of different sizes (1×1, 3×3, and 5×5) to capture spatial information at different scales, including both fine and coarse grain level. The exploitation of the idea of split, transform, and merge by GoogleNet, helped in addressing a problem related to the learning of diverse types of variations present in the same category of images having different resolutions. GoogleNet regulates the computations by adding a bottleneck layer of 1×1 convolutional filter, before employing large size kernels. In addition to it, it used sparse connections (not all the output feature-maps are connected to all the input feature-maps), to overcome the problem of redundant information and reduced cost by omitting feature-maps that were not relevant. Furthermore, connection's density was reduced by using global average pooling at the last layer, instead of using a fully connected layer. These parameter tunings caused a significant decrease in the number of parameters from 138 million to 4 million parameters. Other regulatory factors applied were batch normalization and the use of RmsProp as an optimizer (Dauphin et al. 2015). GoogleNet also introduced the concept of auxiliary learners to speed up the convergence rate. However, the main drawback of the GoogleNet was its heterogeneous topology that needs to be customized from module to module. Another limitation of GoogleNet was a representation bottleneck that drastically reduces the feature space in the next layer and thus sometimes may lead to loss of useful information.

4.2 Depth based CNNs

Deep CNN architectures are based on the assumption that with the increase in depth, the network can better approximate the target function with a number of nonlinear mappings and more enriched feature hierarchies (Bengio 2013). Network depth has played an essential role in the success of supervised training. Theoretical studies have shown that deep networks can represent certain classes of function more efficiently than shallow architectures (Montufar et al. 2014). Csáji represented a universal approximation theorem in 2001, which states that a single hidden layer is sufficient to approximate any function. However, this comes at the cost of exponentially many neurons; thus, it often makes it computationally non-realistic (Csáji 2001). In this regard, Bengio and Delalleau (Delalleau and Bengio 2011) suggested that deeper networks can maintain the expressive power of the network at a reduced cost (Wang and Raj 2017). In 2013, Bengio et al. empirically showed that deep networks are computationally more efficient for complex tasks (Bengio et al. 2013; Nguyen et al. 2018). Inception and VGG, which showed the best performance in 2014-ILSVRC competition, further strengthen the idea that the depth is an essential dimension in regulating learning capacity of the networks (Simonyan and Zisserman 2015; Szegedy et al. 2015, 2016a, b).

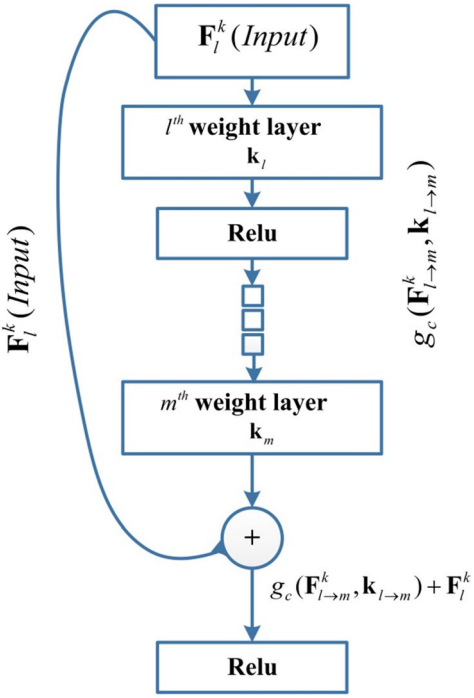

Fig. 7 Residual block as a basic structural unit of ResNet

4.2.1 Highway networks

Based on the intuition that the learning capacity can be improved by increasing the network depth, Srivastava et al. (2015a), proposed a deep CNN, named as Highway Networks. The main problem concerned with deep networks is slow training and convergence speed (Huang et al. 2016b). Highway Networks exploited depth for learning enriched feature representation and introducing a new cross-layer connectivity mechanism (discussed in Sect. 4.3.1) for the successful training of the deep networks. Therefore, Highway Networks are also categorized as multi-path based CNN architectures. Highway Networks with 50-layers showed a better convergence rate than thin but deep architectures (Berg et al. 2010; Morar et al. 2012). Srivastava et al. experimentally showed that the performance of a plain network decreases after adding hidden units beyond 10 layers (Glorot and Bengio 2010). Highway Networks, on the other hand, was shown to converge significantly faster than the plain ones, even with the depth of 900 layers.

4.2.2 ResNet

ResNet was proposed by He et al. (2015a) which is considered as a continuation of deep networks. ResNet revolutionized the CNN architectural race by introducing the concept of residual learning in CNNs and devised an efficient methodology for the training of deep networks. Similar to Highway Networks, it is also placed under the Multi-Path based CNNs; thus, its learning methodology is discussed in Sect. 4.3.2. ResNet proposed 152-layers deep CNN, which won the 2015-ILSVRC competition. The architecture of the

residual block of ResNet is shown in Fig. 7. ResNet, which was 20 and 8 times deeper than AlexNet and VGG, respectively, showed less computational complexity than previously proposed networks (Krizhevsky et al. 2012; Simonyan and Zisserman 2015). He et al. empirically showed that ResNet with 50/101/152 layers has less error on image classification task than 34 layers plain Net. Moreover, ResNet gained a 28% improvement on the famous image recognition benchmark dataset named COCO (Lin et al. 2014). Good performance of ResNet on image recognition and localization tasks showed that representational depth is of central importance for many visual recognition tasks.

4.2.3 Inception-V3, V4 and Inception-ResNet

Inception-V3, V4 and Inception-ResNet, are improved versions of Inception-V1 and V2 (Szegedy et al. 2015, 2016a, b). The idea of Inception-V3 was to reduce the computational cost of deep networks without affecting the generalization. For this purpose, Szegedy et al. (2016b) replaced large size filters (5×5 and 7×7) with small and asymmetric filters (1×7 and 1×5) and used 1×1 convolution as a bottleneck before the large filters. Concurrent placement of 1×1 convolution with a large size filter makes the traditional convolution operation more like a cross-channel correlation. In one of the previous works, Lin et al. exploited the potential of 1×1 filters in NIN architecture (Lin et al. 2013). Szegedy et al. (2016b) intelligently used the same concept. In Inception-V3, 1×1 convolutional operation was used, which maps the input data into 3 or 4 separate spaces that are smaller than the original input space, and then maps all correlations in these smaller 3D spaces, via regular (3×3 or 5×5) convolutions. In Inception-ResNet, Szegedy et al. combined the power of residual learning and inception block (He et al. 2015a; Szegedy et al. 2016a). In doing so, filter concatenation was replaced by the residual connection. Moreover, Szegedy et al. experimentally showed that Inception-V4 with residual connections (Inception-ResNet) has the same generalization power as plain Inception-V4 but with increased depth and width. However, they observed that Inception-ResNet converges more quickly than Inception-V4, which depicts that training with residual connections accelerates the training of Inception networks significantly.

4.3 Multi-path based CNNs

Training of deep networks is a challenging task, and this has been the subject of recent research on deep networks. Deep CNNs generally perform well on complex tasks. However, they may suffer from performance degradation, gradient vanishing, or explosion problems, which are not caused by overfitting but instead by an increase in the depth (Hochreiter 1998; Dong et al. 2016). Vanishing gradient problem not only results in higher test error but also higher training error (Pascanu et al. 2012; Dong et al. 2016; Dauphin et al. 2017). For training deep networks, the concept of multi-path or cross-layer connectivity was proposed (Srivastava et al. 2015a; Larsson et al. 2016; Huang et al. 2017; Kuen et al. 2018). Multiple paths or shortcut connections can systematically connect one layer to another by skipping some intermediate layers to allow the specialized flow of information across the layers (Mao et al. 2016; Tong et al. 2017). Cross-layer connectivity partitions the network into several blocks. These paths also try to solve the vanishing gradient problem by making gradient accessible to lower layers. For this purpose, different types of shortcut

connections are used, such as zero-padded, projection-based, dropout, skip connections, and 1×1 connections, etc.

4.3.1 Highway networks

The increase in depth of a network improves performance mostly for complex problems, but it also makes training of the network difficult. In deep networks, due to a large number of layers, the backpropagation of error may result in small gradient values at lower layers. To solve this problem, in 2015, a new CNN architecture named Highway Networks was proposed based on the idea of cross-layer connectivity (Srivastava et al. 2015a). In Highway Networks, the unimpeded flow of information across layers is enabled by imparting two gating units within a layer (Eq. 5). The idea of a gating mechanism was inspired by Long Short Term Memory (LSTM) based on Recurrent Neural Networks (RNN) (Mikolov et al. 2010; Sundermeyer et al. 2012). The aggregation of information by combining the l^{th} layer and previous $l-j$ layers information creates a regularizing effect, making gradient-based training of very deep networks easy. This cross-layer connectivity enables the training of a network with more than 100 layers, even as deep as 900 layers with a stochastic gradient descent algorithm. Cross-layer connectivity for Highway Network is defined in Eqs. (5 and 6).

$$\mathbf{F}_{l+1}^{k} = g_c(\mathbf{F}_{l}^{k}, \mathbf{k}_l).g_{t_g}(\mathbf{F}_{l}^{k}, {}^{t_g}\mathbf{k}_l).g_{c_g}(\mathbf{F}_{l}^{k}, {}^{c_g}\mathbf{k}_\mathbf{l}) \tag{5}$$

$$g_{c_g}(\mathbf{F}_{l}^{k}, {}^{c_g}\mathbf{k}_l) = 1 - g_{t_g}(\mathbf{F}_{l}^{k}, {}^{t_g}\mathbf{k}_l) \tag{6}$$

In Eq. (5), $g_c(\mathbf{F}_{l}^{k}, \mathbf{k}_l)$ represents the working of the lth hidden layer, whereas t_g and c_g are two gates that decide the flow of information across the layers. When t_g gate is open, $\mathrm{t}_g = 1$ then transformed input is assigned to the next layer. Whereas, when the value of $\mathrm{t}_g = 0$ then c_g gate establishes an effect of information highway and input $\mathbf{F}_{l}^{k}$ of lth layer is directly assigned to the next layer $l+1$ without any transformation.

4.3.2 ResNet

In order to address the problems faced during training of deep networks, ResNet exploited the idea of bypass pathways used in Highway Networks (He et al. 2015a). Mathematical formulation of ResNet is expressed in Eqs. (7, 8 and 9).

$$\mathbf{F}_{m+1}^{k'} = g_c(\mathbf{F}_{l\to m}^{k}, \mathbf{k}_{l\to m}) + \mathbf{F}_{l}^{k} \quad m \geq l \tag{7}$$

$$\mathbf{F}_{m+1}^{k} = g_a(\mathbf{F}_{m+1}^{k'}) \tag{8}$$

$$g_c(\mathbf{F}_{l\to m}^{k}, \mathbf{k}_{l\to m}) = \mathbf{F}_{m+1}^{k'} - \mathbf{F}_{l}^{k} \tag{9}$$

where $g_c(\mathbf{F}_{l\to m}^{k}, \mathbf{k}_{l\to m})$ is a transformed signal, and $\mathbf{F}_{l}^{k}$ is an input of lth layer. In Eq. (7), $\mathbf{k}_{l\to m}$ shows the kth processing unit (kernel), whereas $l \to m$ suggests that the residual block can be consists of one or more than one hidden layers. Original input $\mathbf{F}_{l}^{k}$ is added to transformed signal ($g_c(\mathbf{F}_{l\to m}^{k}, \mathbf{k}_{l\to m})$) through bypass pathway (Eq. 7) and thus results in an

aggregated output $\mathbf{F}_{m+1}^{k'}$, which is assigned to the next layer after applying activation function $g_a(.)$. Whereas, $(\mathbf{F}_{m+1}^{k} - \mathbf{F}_{l}^{k})$, returns a residual information, which is used to perform reference based optimization of weights. The distinct feature of ResNet is reference based residual learning framework. ResNet suggested that residual functions are easy to optimize and can gain accuracy for considerably increased depth.

ResNet introduced shortcut connections within layers to enable cross-layer connectivity; however, these connections are data-independent and parameter-free in comparison to the gates of Highway Networks. In Highway Networks, when a gated shortcut is closed, the layers represent non-residual functions. However, in ResNet, residual information is always passed, and identity shortcuts are never closed. Residual links (shortcut connections) speed up the convergence of deep networks, thus giving ResNet the ability to avoid gradient diminishing problems.

4.3.3 DenseNet

Similar to Highway Networks and ResNet, DenseNet was proposed to solve the vanishing gradient problem (Srivastava et al. 2015a; He et al. 2015a; Huang et al. 2017). The problem with ResNet was that it explicitly preserves information through additive identity transformations due to which many layers may contribute very little or no information. To address this problem, DenseNet used cross-layer connectivity but, in a modified fashion. DenseNet connected each preceding layer to the next coming layer in a feed-forward fashion; thus, feature-maps of all previous layers were used as inputs into all subsequent layers as expressed in Eqs. (10 and 11).

$$\mathbf{F}_{2}^{k} = g_c(I_C, \mathbf{k}_1) \tag{10}$$

$$\mathbf{F}_{l}^{k} = g_k(\mathbf{F}_{1}^{k}, \ldots, \mathbf{F}_{l-1}^{k}) \tag{11}$$

where $\mathbf{F}_{2}^{k}$ and $\mathbf{F}_{l}^{k}$ are the resultant feature-maps of 1st and l-1th transformation layers and $g_k(.)$ is a function, which enables cross-layer connectivity by concatenating proceeding layers information before assigning to new transformation layer l. This establishes $\frac{l(l+1)}{2}$ direct connections in DenseNet, as compared to l connections between a layer and its preceding layer in the traditional CNNs. It imprints the effect of cross-layer depthwise convolutions. As DenseNet concatenates the features of the previous layer instead of adding them, thus, the network may gain the ability to explicitly differentiate between information that is added to the network and information that is preserved. DenseNet has a narrow layer structure; however, it becomes parametrically expensive with an increase in a number of feature-maps. Information flow in the network improves by providing each layer direct access to the gradients through the loss function. Direct admittance to gradient incorporates a regularizing effect, which reduces overfitting on tasks with smaller training sets.

4.4 Width based multi-connection CNNs

During 2012–2015, the focus was mainly on exploiting the power of depth, along with the effectiveness of multi-pass regulatory connections in network regularization (Srivastava et al. 2015a; He et al. 2015a). However, Kawaguchi et al. (2019) reported that the width of the network is also important. Multilayer perceptron gained the advantage of mapping complex functions over perceptron by making parallel use of multiple processing units within a layer. This suggests that width is an essential parameter in defining principles of

learning along with depth. Lu et al. (2017a, b), and Hanin and Sellke (2017) have recently shown that NNs with ReLU activation function have to be wide enough to hold universal approximation property along with an increase in depth (Hanin and Sellke 2017). Moreover, a class of continuous functions on a compact set cannot be arbitrarily well approximated by an arbitrarily deep network, if the maximum width of the network is not larger than the input dimension (Lu et al. 2017b; Nguyen et al. 2018). Although, stacking of multiple layers (increasing depth) may learn diverse feature representations, but may not necessarily increase the learning power of the NN. One major problem linked with deep architectures is that some layers or processing units may not learn useful features. To tackle this problem, the focus of research shifted from deep and narrow architecture towards thin and wide architectures.

4.4.1 Wide ResNet

It is concerned that the main drawback associated with deep residual networks is the feature reuse problem in which some feature transformations or blocks may contribute very little to learning (Srivastava et al. 2015b). This problem was addressed by Wide ResNet (Zagoruyko and Komodakis 2016). Zagoruyko and Komodakis suggested that the main learning potential of deep residual networks is due to the residual units, whereas depth has a supplementary effect. Wide ResNet exploited the power of the residual blocks by making ResNet wide rather than deep (He et al. 2015a). Wide ResNet increased the width by introducing an additional factor k, which controls the width of the network. Wide ResNet showed that the widening of the layers might provide a more effective way of a performance improvement than by making the residual networks deep.

Deep networks improved representational capacity, but they have some demerits such as time-intensive training, feature reuse, and gradient vanishing and exploding problem. He et al. (2015a) addressed feature reuse problem by incorporating dropout in residual blocks to regularize network effectively. Similarly, Huang et al. (2016a) introduced the concept of stochastic depth by exploiting dropouts to solve vanishing gradient and slow learning problems. It was observed that even fraction improvement in performance might require the addition of many new layers. However, Zagoruyko and Komodakis (2016), empirically showed that though Wide ResNet was twice in a number of parameters as compared to ResNet, but can be trained in a better way than the deep networks (Zagoruyko and Komodakis 2016). Wide ResNet was based on the observation that almost all architectures before residual networks, including the most successful Inception and VGG, were wide as compared to ResNet. In Wide ResNet, learning is made effective by adding a dropout in between the convolutional layers rather than inside a residual block.

4.4.2 Pyramidal net

In earlier deep CNN architectures such as AlexNet, VGG, and ResNet, due to the deep stacking of multiple convolutional layers, depth of feature-maps increases in subsequent layers. However, the spatial dimension decreases, as each convolutional layer or block is followed by a sub-sampling layer (Krizhevsky et al. 2012; Simonyan and Zisserman 2015; He et al. 2015a). Therefore, Han et al. (2017) argued that in deep CNNs, a drastic increase in the feature-map depth and, at the same time, the loss of spatial information limits the learning ability of CNN. ResNet has shown remarkable results for image classification

problems. However, in ResNet, the deletion of a residual block, where the dimension of both spatial and feature-map (channel) varies (feature-map depth increases, while spatial dimension decreases), generally deteriorates performance. In this regard, stochastic ResNet improved the performance by reducing information loss associated with the dropping of the residual unit (Huang et al. 2016a). To increase the learning ability of ResNet, Han et al. (2017) proposed the Pyramidal Net. In contrast to drastic decrease in spatial width with an increase in depth by ResNet, Pyramidal Net increases the width gradually per residual unit. This strategy enables pyramidal Net to cover all possible locations instead of maintaining the same spatial dimension within each residual block until down-sampling occurs. Because of a gradual increase in the depth of features map in a top-down fashion, it was named as pyramidal Net. In pyramidal network, depth of feature-maps is regulated by factor l, and is computed using Eq. (12).

$$d_l = \begin{cases} 16 & if\ l = 1, \\ \left\lfloor d_{l-1} + \frac{\lambda}{n} \right\rfloor & if 2 \leq l \leq n+1 \end{cases} \tag{12}$$

where d_l denotes the dimensions of lth residual block and n describes the number of the residual block, whereas λ is a step size and $\frac{\lambda}{n}$ regulates the increase in depth. The depth regulating factor tries to distribute the burden of increase in depth of feature-maps. Residual connections were inserted in between the layers by using zero-padded identity mapping. The advantage of zero-padded identity mapping is that it needs less number of parameters as compared to the projection-based shortcut connection, hence may result in better generalization (Wang et al. 2019). Pyramidal Net uses two different approaches for the widening of the network, including addition and multiplication based widening. The difference between the two types of widening is that additive pyramidal structure increases linearly, whereas multiplicative one increases geometrically (Ioffe and Szegedy 2015; Xu et al.

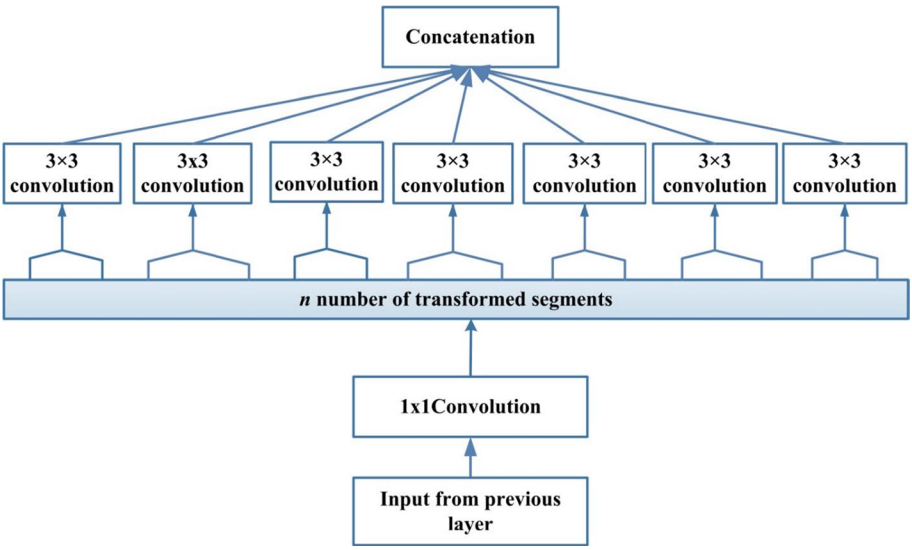

Fig. 8 Xception building block and its n sets of transformation

2015a). However, a major problem with Pyramidal Net is that with the increase in width, a quadratic times increase in both space and time occurs.

4.4.3 Xception

Xception can be considered as an extreme Inception architecture, which exploits the idea of depthwise separable convolution (Chollet 2017). Xception modified the original inception block by making it wider and replacing the different spatial dimensions (1×1, 5×5, 3×3) with a single dimension (3×3) followed by a 1×1 convolution to regulate computational complexity.

The Architecture of the Xception block is shown in Fig. 8. Xception makes the network computationally efficient by decoupling spatial and feature-map (channel) correlation, which is mathematically expressed in Eqs. (13 and 14). It works by first mapping the convolved output to low dimensional embeddings using 1×1 convolutions. It then spatially transforms it n times, where n is a width defining cardinality, which determines the number of transformations.

$$f^k_{l+1}(p,q) = \sum_{x,y} f^k_l(x,y).e^k_l(u,v) \tag{13}$$

$$\mathbf{F}^k_{l+2} = g_c(\mathbf{F}^k_{l+1}, \mathbf{k}_{l+1}) \tag{14}$$

In Eq. (14), $\mathbf{k}_l$ is a kth kernel of lth layer having depth one, which is spatially convolved across kth feature-map $\mathbf{F}^k_l$, where (x,y) and (u,v) show the spatial indices of feature-map and kernel respectively. In depthwise separable convolution, it is to be noted that number of kernels K is equal to number of input feature-maps contrary to conventional convolutional layer where number of kernels are independent of previous layer feature-maps. Whereas $\mathbf{k}_{l+1}$ is kth kernel of (1×1) spatial dimension for $l+1$th layer, which performs depthwise convolution across output feature-maps $[\mathbf{F}^1_{l+1}, \ldots, \mathbf{F}^k_{l+1}, \ldots, \mathbf{F}^K_{l+1}]$ of lth layer, used as input of $l+1$th layer.

Xception makes computation easy by separately convolving each feature-map across spatial axes, which is followed by pointwise convolution (1×1 convolutions) to perform cross-channel correlation. In conventional CNN architectures; convolutional operation uses only one transformation segment, inception block uses three transformation segments, whereas in Xception number of transformation segments is equal to the number of feature-maps. Although the transformation strategy adopted by Xception does not reduce the number of parameters, it makes learning more efficient and results in improved performance.

4.4.4 ResNeXt

ResNeXt, also known as Aggregated Residual Transform Network, is an improvement over the Inception Network (Xie et al. 2017). Xie et al. exploited the concept of the split, transform, and merge in a powerful but simple way by introducing a new term; cardinality (Szegedy et al. 2015). Cardinality is an additional dimension, which refers to the size of the set of transformations (Han et al. 2018; Sharma and Muttoo 2018). The Inception network

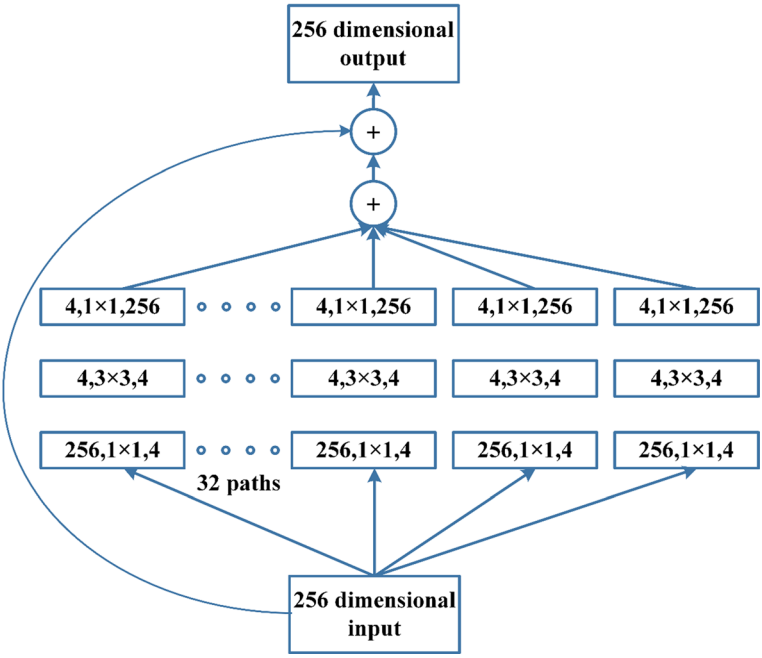

Fig. 9 ResNeXt building block showing the different paths of transformation

has not only improved the learning capability of conventional CNNs, but it also makes a network resource-efficient. However, due to the use of diverse spatial embedding's (such as the use of 3×3, 5×5, and 1×1 filter) in the transformation branch, each layer needs to be customized separately. ResNeXt utilized the deep homogenous topology of VGG and simplified GoogleNet architecture by fixing spatial resolution to 3×3 filters within the split, transform, and merge block. Whereas, it used residual learning to improve the convergence of deep and wide network (Simonyan and Zisserman 2015; Szegedy et al. 2015; He et al. 2015a). The building block for ResNeXt is shown in Fig. 9. ResNeXt used multiple transformations within a split, transform and merge block and defined these transformations in terms of cardinality. Xie et al. (2017) showed that an increase in cardinality significantly improves performance. The complexity of ResNeXt was regulated by applying low embedding's (1×1 filters) before 3×3 convolution, whereas training was optimized by using skip connections (Larsson et al. 2016).

4.4.5 Inception family

Inception family of CNNs also comes under the class of width based methods (Szegedy et al. 2015, 2016a, b). In Inception networks, within a layer, varying sizes of the filters were used, which increased the output of the intermediate layers. The use of the different sizes of filters helps capture the diversity in high-level features. Salient characteristics of the Inception family are discussed in Sects. 4.1.5 and 4.2.3.

4.5 Feature-Map (Channel$_{FMap}$) Exploitation based CNNs

CNN became popular for MV tasks because of its hierarchical learning and automatic feature extraction ability (LeCun et al. 2010). Feature selection plays a vital role in determining the performance of classification, segmentation, and detection modules. In CNN, features are dynamically selected by tuning the weights associated with a kernel also known as mask. Also, multiple stages of feature extraction are used, which can extract diverse types of features (known as feature-maps or channels in CNN). However, some of the feature-maps impart little or no role in object discrimination (Hu et al. 2018a). Enormous feature sets may create an effect of noise and thus lead to over-fitting of the network. This suggests that apart from network engineering, selection of feature-maps can play an important role in improving the generalization of the network. In this section, feature-maps and channels will be interchangeably used as many researchers have used the word channels for the feature-maps.

4.5.1 Squeeze and excitation network

Squeeze and Excitation Network (SE-Network) was reported by Hu et al. (2018a). They proposed a new block for the selection of feature-maps (commonly known as channels) relevant to object discrimination. This new block was named as SE-block (shown in Fig. 10), which suppresses the less important feature-maps, but gives high weightage to the class specifying feature-maps. SE-Network reported a record decrease in error on the ImageNet dataset. SE-block is a processing unit that is designed generically and therefore, can be added in any CNN architecture before the convolution layer. The working of this block consists of two operations; squeeze and excitation. Convolution kernel captures information locally, but it ignores the contextual relation of features (correlation) that are outside of this receptive field (LeCun et al. 2015). Squeeze operation is performed to get a global view of feature-maps. The squeeze block generates feature-map wise statistics (also known as feature-map motifs or descriptors) by suppressing spatial information of the convolved input. As global average pooling has the potential to learn the extent of target object effectively (Lin et al. 2013; Zhou et al. 2016), therefore, it is employed by the squeeze operation $g_{sq}(.)$ using the following Eq. (15):

$$s_l^k = g_{sq}(\mathbf{F}_l^k) = \frac{1}{P \times Q} \sum_{p,q} f_l^k(p, q) \tag{15}$$

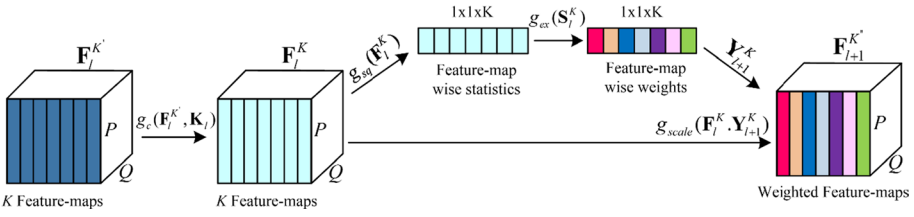

Fig. 10 Squeeze and Excitation block showing the computation of masks for the recalibration of feature-maps that are commonly known as channels in literature

where s_l^k represents a feature descriptor for kth feature-map of lth layer, and $P \times Q$ defines the spatial dimension of feature-map $\mathbf{F}_l^k$. Whereas output of squeeze operation $\mathbf{S}_l^K = \left[s_l^1, \ldots, s_l^K\right]$ for K number of convolved feature-maps for lth layer is assigned to the excitation operation $g_{ex}(.)$, which models motif-wise interdependencies by exploiting gating mechanism. Excitation operation assigns weights to feature-maps using two layer feed forward NN, which is mathematically expressed in Eq. (16).

$$y_{l+1}^k = g_{ex}(\mathbf{S}_l^k) = g_{s_g}(\mathbf{w}_2, g_t(\mathbf{S}_l^k, \mathbf{w}_1)) \tag{16}$$

In Eq. (16), y_{l+1}^k denotes weightage for input feature-map $\mathbf{F}_{l+1}^k$ of next layer $(l+1)$, where $g_t(.)$ and $g_{s_g}(.)$ apply the ReLU based non-linear transformation and sigmoid gate, respectively. Similarly, $\mathbf{Y}_{l+1}^K = \left[y_{l+1}^1, \ldots, y_{l+1}^K\right]$ shows the weightage for K number of convolved feature-maps that are used to rescale them before assigning to the $l+1$th layer. In excitation operation, $\mathbf{w}_1$ and $\mathbf{w}_2$ both are used as transformation weight vectors and regulating factors to limit the model complexity and aid the generalization (LeCun 2007; Xu et al. 2015a). The output of the first hidden transformation in NN is preceded by the ReLU activation function, which inculcates non-linearity in motif responses. The gating mechanism is exploited in SE-block using the sigmoid activation function, which models the non-linear responses of the feature-maps and assigns a weight based on feature-map relevance (Zheng et al. 2017). SE-block adaptively recalibrates the feature-maps of each layer by multiplying convolved input with the motif responses.

4.5.2 Competitive squeeze and excitation networks

Competitive Inner-Imaging Squeeze and Excitation for Residual Network also known as CMPE-SE Network was proposed by Hu et al. (2018b). Hu et al. (2018a) used the idea of SE-block to improve the learning of deep residual networks. SE-Network recalibrates the feature-maps based upon their contribution in class discrimination. However, the main concern with SE-Net is that in ResNet, it only considers the residual information for determining the weight of each feature-map (Hu et al. 2018a). This minimizes the impact of SE-block and makes ResNet information redundant. Hu et al. addressed this problem by generating feature-map wise motifs (statistics) from both residual and identity mapping based feature-maps. In this regard, global representation of feature-maps is generated using global average pooling operation (Eq. 17), whereas relevance of feature-maps is estimated by establishing competition between feature descriptors of residual and identity mappings. This phenomena is termed as inner imaging (Hu et al. 2018b). CMPE-SE block not only models the relationship between residual feature-maps but also maps their relation with identity feature-map. The mathematical expression for CMPE-SE block is represented using the following equation:

$$\mathbf{S}_l^k, \mathbf{S}_{m+1}^k = g_{sq}(\mathbf{F}_l^k), g_{sq}(\mathbf{F}_{m+1}^{k'}) \tag{17}$$

$$\mathbf{Y}_{m+1}^k = g_{ex}(g_k(\mathbf{S}_l^k, \mathbf{S}_{m+1}^k)) \tag{18}$$

$$\mathbf{F}_{m+1}^k = \mathbf{Y}_{m+1}^k . \mathbf{F}_{m+1}^{k'} \tag{19}$$

where $\mathbf{F}_l^K$ and $\mathbf{F}_{m+1}^{K'}$ are the identity and residual mapping of input $\mathbf{F}_l^K$ respectively. SE block is implemented by applying squeeze operation $g_{sq}(.)$ both on residual and the identity feature-maps and their receptive output is used as joint input of excitation operation $g_{ex}(.)$. Whereas $g_k(.)$ represents the concatenation operation. The output masks of excitation operation (Eq. 18) are multiplied with residual information (Eq. 19) to rebuild each feature-map importance. The backpropagation algorithm thus tries to optimize the competition between identity and residual feature-maps and the relationship between all feature-maps in the residual block.

4.6 Channel$_{(Input)}$ exploitation based CNNs

Image representation plays an important role in determining the performance of the image processing algorithms, including both conventional and deep learning algorithms. A good representation of the image is one that can define the salient features of an image from a compact code. In MV tasks, various types of conventional filters are applied to extract different levels of information for a single type of image (Lowe 2004; Dollár et al. 2009). These diverse representations are then used as an input of the model to improve performance (Do and Vetterli 2005; Oquab et al. 2014). Now CNN is a compelling feature learner that can automatically extract discriminating features depending upon the problem (Yang et al. 2019). However, the learning of CNN relies on input representation. The lack of diversity and the absence of class discernable information in the input may affect CNN's performance as a discriminator. For this purpose, the concept of channel boosting (input channel dimension) using auxiliary learners is introduced in CNNs to boost the representation of the network (Khan et al. 2018a).

4.6.1 Channel boosted CNN using TL

Khan et al. (2018a) proposed a new CNN architecture named as Channel boosted CNN (CB-CNN) based on the idea of boosting the number of input channels for improving the representational capacity of the network. The Block diagram of CB-CNN is shown in Fig. 11. Channel boosting is performed by artificially creating extra channels (known as auxiliary channels) through auxiliary deep generative models and then exploiting it through the deep discriminative models. CB-CNN is mathematically expressed in Eqs. (20 and 21).

$$\mathbf{I}_B = g_k\left(\mathbf{I}_C, \left[\mathbf{A}_1, \ldots, \mathbf{A}_M\right]\right) \tag{20}$$

$$\mathbf{F}_l^k = g_c(\mathbf{I}_B, \mathbf{k}_l) \tag{21}$$

In Eq. (20), $\mathbf{I}_C$ represents the original input channels, where $\mathbf{A}_M$ is an artificial channel generated by Mth auxiliary learner. Whereas $g_k(.)$ is used as a combiner function that concatenates the original input channels with auxiliary channels to generates the channel boosted input $\mathbf{I}_B$ for the discriminator. Equation (21) shows the kth resultant feature-map $\mathbf{F}_l^k$, which is generated by convolving the boosted input $\mathbf{I}_B$ with kernel $\mathbf{k}_l$ of lth layer.

Bengio et al. (2013), empirically showed that data representation plays an important role in determining the performance of a classifier, as different representations may present different aspects of information. For improving the representation of the data, Khan et al. exploited the power of TL and deep generative learners (Qiang Yang et al. 2008; Vincent et al. 2008; Hamel and Eck 2010). Generative learners attempt to characterize the data

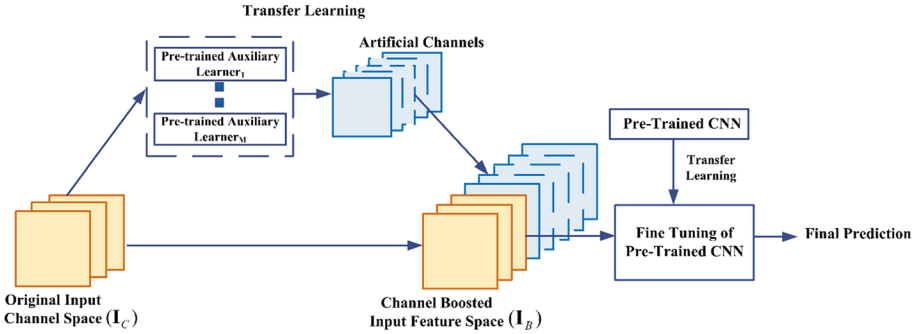

Fig. 11 Basic architecture of CB-CNN showing the deep auxiliary learners for creating artificial channels

Table 4 Results of CNN and CB-CNN on mitosis dataset

CNN architecture	F-score
26 layers deep CNN	0.47
26 layers deep CB-CNN	0.53
VGG	0.55
CB-VGG	0.71
ResNet	0.44
CB-ResNet	0.54

generating distribution during the learning phase. In CB-CNN, AEs are used as the generative learners to learn explanatory factors of variation behind the data. The concept of inductive TL is used in a novel way to build a boosted input representation by augmenting learned distribution of the input data with the original channel space (input channels). CB-CNN encodes channel-boosting phase into a generic block, which is inserted at the start of a deep network. CB-CNN provides the concept that TL can be used at both generation and discrimination stages. The significance of the study is that multi-deep learners are used, where generative learning models are used as auxiliary learners. These leaners enhance the representational capacity of deep CNN based discriminator. Although the potential of the channel boosting was only evaluated by inserting a boosting block at the start, however, Khan et al. suggested that this idea can be extended by providing auxiliary channels at any layer in the deep architecture. CB-CNN has also been evaluated on the medical image dataset (Aziz et al. 2020), where it has shown improved results, as shown in Table 4.

4.7 Attention based CNNs

Different levels of abstractions have an important role in defining the discrimination power of the NN. In addition to learning of multiple hierarchies of abstractions, focusing on features relevant to the context also plays a significant role in image localization and recognition. In the human visual system, this phenomenon is referred to as attention. Humans view the scene in a succession of partial glimpses and pay attention to context-relevant parts. This process not only serves to focus selected regions but also deduces different interpretations of objects at that location and thus helps in capturing visual structure in a better way.

A more or less similar kind of interpretability is added into RNN and LSTM (Mikolov et al. 2010; Sundermeyer et al. 2012). RNN and LSTM networks exploit attention modules for the generation of sequential data, and the new samples are weighted based on their occurrence in previous iterations. The concept of attention was incorporated into CNN, by various researchers to improve representation and overcome the computational limits. This idea of attention also helps in making CNN intelligent enough to recognize objects even from cluttered backgrounds and complex scenes.

4.7.1 Residual attention neural network

Wang et al. (2017a) proposed a Residual Attention Network (RAN) to improve the feature representation of the network. The motivation behind the incorporation of attention in CNN was to make a network capable of learning object aware features. RAN is a feed-forward CNN, which was built by stacking residual blocks with attention module. The attention module is branched off into trunk and mask branches that adopt bottom-up, top-down learning strategy. The assembly of two different learning strategies into the attention module enables fast feed-forward processing and top-down attention feedback in a single feed-forward process. The bottom-up feed-forward structure produces low-resolution feature-maps with strong semantic information. Whereas, top-down architecture produces dense features to make an inference of each pixel.

In the previously proposed studies, a top-down, bottom-up learning strategy was used by Restricted Boltzmann Machines (Salakhutdinov and Larochelle 2010). Similarly, Goh et al. exploited the top-down attention mechanism as a regularizing factor in Deep Boltzmann Machine during the reconstruction phase of the training. The top-down learning strategy globally optimizes the network in such a way that it gradually output the maps to input during the learning process (Hinton et al. 2006; Salakhutdinov and Larochelle 2010; Goh et al. 2013). The attention module in RAN generates object aware soft mask $g_{sm}(.)$ at each layer for input feature-map $\mathbf{F}_l^K$ (Xu et al. 2015b). Soft mask $g_{sm}(.)$ assigns attention towards object using Eq. (22) by recalibrating trunk branch $g_{tm}(\mathbf{F}_l^K)$ output and thus, behaves like a control gate for every neuron output.

$$g_{am}(\mathbf{F}_l^K) = g_{sm}(\mathbf{F}_l^K).g_{tm}(\mathbf{F}_l^K) \tag{22}$$

In one of the previous studies, Transformation network (Jaderberg et al. 2015; Li et al. 2018) also exploited the idea of attention in a simple way by incorporating it with convolution block, but the main problem was that attention modules in Transformation network are fixed and cannot adapt to changing circumstances. RAN was made efficient towards recognition of cluttered, complex, and noisy images by stacking multiple attention modules. Hierarchical organization of RAN endowed the ability to adaptively assign weight to each feature-map based on their relevance in the layers (Wang et al. 2017a). Learning of deep hierarchical structure was supported through residual units. Moreover, three different levels of attention: mixed, channel, and spatial attention were incorporated, thus leveraging the capability to capture object-aware features at different levels (Wang et al. 2017a).

4.7.2 Convolutional block attention module

The significance of attention mechanism and feature-map exploitation is validated through RAN and SE-Network (Wang et al. 2017a; Hu et al. 2018a). In this regard, Woo et al. (2018) came up with new attention-based CNN, named as Convolutional Block Attention Module (CBAM). CBAM is simple in design and similar to SE-Network. SE-Network only considers the contribution of feature-maps in image classification, but it ignores the spatial locality of the object in images. The spatial location of the object has a vital role in object detection.

CBAM infers attention maps sequentially by first applying feature-map (channel) attention and then spatial attention, to find the refined feature-maps. In literature, generally, 1×1 convolution and pooling operations are used for spatial attention. Woo et al. showed that the pooling of features along the spatial axis generates an efficient feature descriptor. CBAM concatenates average pooling operation with max-pooling, which generates a strong spatial attention map. Likewise, feature-map statistics were modeled using a combination of max-pooling and global average-pooling operation. Woo et al. showed that max-pooling could provide the clue about distinctive object features, whereas the use of global average pooling returns suboptimal inference of feature-map attention. The exploitation of both average-pooling and max-pooling improves the representational power of the network. These refined feature-maps not only focus on the important part but also increase the representational power of the selected feature-maps. Woo et al. empirically showed that the formulation of a 3D attention map via the serial learning process helps in the reduction of the parameters as well as computational cost. Due to the simplicity of CBAM, it can be integrated easily with any CNN architecture.

4.7.3 Concurrent spatial and channel excitation mechanism

Roy et al. (2018) extended the work of Hu et al. (2018a) by incorporating the effect of spatial information in combination with feature-map (channel) information to make it applicable to segmentation tasks. They introduced three different modules: (1) squeezing spatially and exciting feature-map information (cSE), (2) squeezing feature-map and exciting spatial information (sSE), and (3) concurrent squeeze and excitation of spatial and feature-map information (scSE). In this work, AE based convolutional NN was used for segmentation, whereas proposed modules were inserted after the encoder and decoder layers. In the cSE module, the same concept as that of SE-block is exploited. In this module, the scaling factor is derived based on the combination of feature-maps used for object detection. As spatial information has an important role in segmentation, therefore in the sSE module, the spatial locality has been given more importance than feature-map information. For this purpose, different combinations of feature-maps are selected and exploited spatially to use them for segmentation. In the last module, scSE, attention to each feature-map, is assigned by deriving scaling factor both from spatial and feature-map information and thus to highlight the object-specific feature-maps [117] selectively (Roy et al. 2018).

5 Applications of CNNs

CNNs have been successfully applied to different ML related tasks, namely object detection, recognition, classification, regression, segmentation, etc., (Batmaz et al. 2019; Chouhan and Khan 2019; Wahab et al. 2019). However, CNN generally needs a large amount of data for learning. All of the areas mentioned earlier in which CNN has shown tremendous success have relatively sufficient labeled data, such as traffic sign recognition, segmentation of medical images, and the detection of faces, text, pedestrians, and humans in natural images. Some of the interesting applications of CNN are discussed below.

5.1 CNN based computer vision and related applications

Computer vision (CV) focuses on developing an artificial system that can process visual data, including images and videos and can effectively understand, and extract useful information from it. CV encompasses a number of application areas such as face recognition, pose estimation, activity recognition, etc.

Face recognition is one of the difficult tasks in the CV. Face recognition systems have to cope with variations such as caused by illumination, change in pose, and different facial expressions. Farfade et al. (2015) proposed deep CNN for detecting faces from different poses as well as from occluded faces (Farfade et al. 2015). In another work, Zhang et al. (2016) performed face detection using a new type of multitasking cascaded CNN. Zhang's technique showed good results when compared to state-of-the-art techniques (Li et al. 2015; Ranjan et al. 2015; Yang et al. 2015).

Human pose estimation is one of the challenging tasks related to CV because of the high variability in body pose. Li et al. (2014) proposed a heterogeneous deep CNN based pose estimation related technique. In Li's technique, empirical results have shown that the hidden neurons can learn the localized part of the body. Similarly, another cascade based CNN technique is proposed by Bulat and Tzimiropoulos (2016). In their cascaded architecture, first heat maps are detected, whereas, in the second phase, regression is performed on the detected heat maps.

Action recognition is one of the important areas of activity recognition. The difficulties in developing an action recognition system are to solve the translations and distortions of features in different patterns, which belong to the same action class. Earlier approaches involved the construction of motion history images, the use of Hidden Markov Models, action sketch generation, etc. Recently, Wang et al. (2017b) proposed a three dimensional CNN architecture in combination with LSTM for recognizing different actions from video frames. Experimental results have shown that Wang's technique outperforms other activity recognition based techniques (Wang and Schmid 2013; Simonyan and Zisserman 2014; Donahue et al. 2015; Sun et al. 2015; Tran et al. 2015). Similarly, another three dimensional CNN based action recognition system is proposed by Ji et al. (2010). In Ji's work, three-dimensional CNN is used to extract features from multiple channels of input frames. The final action recognition based model is developed on combined extracted feature space. The proposed three dimensional CNN model is trained in a supervised way and can perform activity recognition in real-world applications.

5.2 CNN based natural language processing

Natural Language Processing (NLP) converts language into a presentation that can easily be exploited by any computer. Although RNNs are very suitable for NLP applications, however, CNNs have also been utilized in NLP based applications such as language modeling, and analysis, etc. Especially, language modeling or sentence molding has taken a twist after the introduction of CNN as a new representation learning algorithm. Sentence modeling is performed to know semantics of the sentences and thus offer new and appealing applications according to customer requirements. Traditional methods of information retrieval analyze data, based on words or features, but ignore the core of the sentence. Kalchbrenner et al. (2014) proposed a dynamic CNN and dynamic *k-max* pooling during training. This approach finds the relations between words without taking into account any external source like parser or vocabulary (Kalchbrenner et al. 2014). In a similar way, Collobert and Weston (2008) proposed CNN based architecture that can perform various MLP related tasks at the same time as chunking, language modeling, recognizing name-entity, and role modeling related to semantics. In another work, Hu et al. (2011) proposed a generic CNN based architecture that performs matching between two sentences and thus can be applied to different languages.

5.3 CNN based object detection and segmentation

Object detection focuses on identifying different objects in images. Recently, R-CNN has been widely used for object detection. Ren et al. (2015) proposed an improvement over R-CNN named as fast R-CNN for object detection. In their work, a fully connected convolutional neural network is used to extract feature space that can simultaneously detect the boundary and score of objects located at different positions. Similarly, Dai et al. (2016) proposed region-based object detection using fully connected CNN. In Dai's work, results are reported on the PASCAL VOC image dataset. Another object detection technique is reported by Gidaris and Komodakis (2015), which is based on multi-region based deep CNN that helps to learn the semantic aware features. In Gidaris's approach, objects are detected with high accuracy on PASCAL VOC 2007 and 2012 dataset. Recently, AE based CNN architectures have shown success in segmentation tasks. In this regard, various interesting CNN architectures have been reported for both semantic and instance-based segmentation tasks such as FCN, SegNet, Mask R-CNN, U-Net etc., (Ronneberger et al. 2015; Badrinarayanan et al. 2017; He et al. 2017; Zhang et al. 2018b).

5.4 CNN based image classification

CNN has been widely used for image classification (Levi and Hassner 2009; Long et al. 2012; Sermanet et al. 2012). One of the primary applications of CNN is in medical images, especially for the diagnosis of cancer using histopathological images (Cireşan et al. 2013). Recently, Spanhol et al. (2016a, b) used CNN for the diagnosis of breast cancer images, and results are compared against a network trained on a dataset containing handcrafted descriptors. Another recently proposed CNN based technique for breast cancer diagnosis is developed by Wahab et al. (2017). In Wahab's work, two phases are involved. In the first phase, hard non-mitosis examples are identified. In the second

phase, data augmentation is performed to cope with the class skewness problem. Similarly, Cireşan et al. (2012a, b) used the German benchmark dataset related to a traffic sign signal. They designed CNN based architecture that performed traffic sign classification related task with a good recognition rate.

5.5 CNN based speech recognition

Deep CNN is mostly considered as the best option to deal with image processing applications, however; recent studies have shown that it also performs well on speech recognition tasks. Abdel-Hamid et al. (2012) reported a CNN based speaker-independent speech recognition system. Experimental results showed a ten percent reduction in error rate in comparison to the earlier reported methods (Dahl et al. 2010; Mohamed et al. 2012). In another work, various CNN architectures, which are either based on the full or limited number of weight sharing within the convolution layer, are explored (Abdel-Hamid et al. 2013). Furthermore, the performance of CNN is also evaluated after the initialization of the network using the pre-training phase (Mohamed et al. 2012). Experimental results showed that almost all of the explored architectures yield good performance on phone and vocabulary recognition related tasks. Nowadays, the utilization of CNNs for speech emotion recognition is also gaining attention. Huang et al. used CNN in combination with LSTM for recognizing emotions of speech. In Huang's approach, CNN was trained both on verbal and nonverbal segments of speech, and CNN learned features were used by LSTM for recognizing speech emotions (Huang et al. 2019).

5.6 CNN based video processing

In video processing techniques, temporal and spatial information from videos is exploited. Many researchers have used CNN to solve various video processing-related problems (Tong et al. 2015; Frizzi et al. 2016; Shi et al. 2017; Ullah et al. 2017; Wang et al. 2017b). Tong et al. proposed CNN based short boundary detection system. In Tong's approach, TAGs are generated using CNN (Tong et al. 2015). During the experiment, the merging of TAGs against one-shot is performed to annotate video against that shot. Similarly, Wang et al. (2017b) used 3-D CNN along with LSTM to recognize action within the video. In another technique, Frizzi et al. (2016) used CNN for detecting smoke and fire within the video. In Frizzi's approach, CNN architecture not only extracts salient features but also performs the classification task. In the field of action recognition, the gathering of spatial and temporal information is considered as a tedious task. In order to overcome the deficiencies of traditional feature descriptors, Shi et al. (2017) proposed a three stream-based structure, which is capable of extracting spatial–temporal features along with short and long term motion within the video. Similarly, in another technique, CNN, in combination with bi-directional LSTM, is used for recognizing action from the video (Ullah et al. 2017). Their approach comprises of two phases. In the first phase, features are extracted from the sixth frame of the videos. In the second phase, sequential information between features of the frame is exploited using the bi-directional LSTM framework.

5.7 CNN for low resolution images

In the field of ML, different researchers have used CNN based image enhancement techniques for enhancing the resolution of the images (Kawashima et al. 2017; Chevalier et al.

2015; Peng et al. 2016). Peng et al. (2016) used deep CNN based approach, which categorizes the objects in images having low resolution. Similarly, Chevalier et al. (2015) introduced LR-CNN for low-resolution image classification. Another, the deep learning based technique is reported by Kawashima et al. (2017), in which convolutional layers, along with a layer of LSTM is used to recognize action from thermal images of low resolution.

5.8 CNN for resource limited systems

Despite CNN's high computational cost, it has been successfully utilized in developing different ML based embedded systems (Bettoni et al. 2017; Lee et al. 2017; Xie et al. 2018). Lee et al. (2017) developed the number plate recognition system, which is capable of quickly recognizing the number on the license plate. In Lee's technique, the deep learning based embedded recognition system comprises of simple AlexNet architecture. In order to address power efficiency and portability for embedded platforms, Bettoni et al. (2017) implemented CNN on the FPGA platform. In another technique, the FPGA embedded platform is used for efficiently performing different CNN based ML tasks (Xie et al. 2018). Similarly, resource-limited CNN architectures such as MobileNet, ShuffleNet, ANTNets, etc. are highly applicable for mobile devices (Howard et al. 2017; Zhang et al. 2018a; Xiong et al. 2019). Shakeel et al. developed a real-time based driver drowsiness detection application for smart devices such as android phones. They used MobileNet architecture in combination with SSD to exploit the benefit of the lightweight architecture of MobileNet that can be easily deployed on resource-constrained hardware and can learn enriched representation from the incoming video (Shakeel et al. 2019).

5.9 CNN for 1D-data

CNN has not only shown good performance on images but also on 1D-data. The use of 1D-CNN as compared to other ML methods is becoming popular because of its good feature extraction ability. Vinayakumar et al. (2017) used 1D-CNN in combination with RNN, LSTM, and gated recurrent units for intrusion detection in network traffic. They evaluated the performance of the proposed models on the KDDCup 99 dataset consisting of network traffic of TCP/IP packets and showed that CNN significantly surpasses the performance of classical ML models. Abdeljaber et al. (2017) showed that 1D-CNN could be used for real-time structural damage detection problem. 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. Yildirim et al. developed 16 layers deep 1D-CNN based application for mobile devices and a cloud-based environment for detecting cardiac irregularity in ECG signals. They achieved 91.33% accuracy on the MIT-BIH Arrhythmia database (Yıldırım et al. 2018).

6 CNN challenges

Deep CNNs have achieved good performance on data that either is of the time series nature or follows a grid-like topology. However, there are also some other challenges, where deep CNN architectures have been put to tasks. Major challenges associated with different CNN architectures are mentioned in Tables 5, 6, 7, 8, 9, 10, 11. The different researchers related

Table 5 Major challenges associated with implementation of spatial exploitation based CNN architectures

Architecture	Strength	Gaps
Spatial exploitation	As convolutional operation considers the neighborhood (correlation) of input pixels, therefore different levels of correlation can be explored by using different filter sizes	
LeNet	Exploited spatial correlation to reduce the computation and number of parameters Automatic learning of feature hierarchies	Poor scaling to diverse classes of images Large size filters Low level feature extraction
AlexNet	Low, mid and high-level feature extraction using large and small size filters on initial (5×5 and 11×11) and last layers (3×3) Give an idea of deep and wide CNN architecture Introduced regularization in CNN Started parallel use of GPUs as an accelerator to deal with complex architectures	Inactive neurons in the first and second layers Aliasing artifacts in the learned feature-maps due to large filter size
ZfNet	Introduced the idea of parameter tuning by visualizing the output of intermediate layers Reduced both the filter size and stride in the first two layers of AlexNet	Extra information processing is required for visualization
VGG	Proposed an idea of effective receptive field Gave the idea of simple and homogenous topology	Use of computationally expensive fully connected layers
GoogLeNet	Introduced the idea of using Mutiscale Filters within the layers Gave a new idea of split, transform, and merge Reduce the number of parameters by using bottleneck layer, global average-pooling at last layer and Sparse Connections Use of auxiliary classifiers to improve the convergence rate	Tedious parameter customization due to heterogeneous topology May lose the useful information due to representational bottleneck

Table 6 Major challenges associated with implementation of depth based CNN architectures

Architecture	Strength	Gaps
Depth	With the increase in depth, the network can better approximate the target function with a number of nonlinear mappings and improved feature representations. Main challenge faced by deep architectures is the problem of vanishing gradient and negative learning	
Inception-V3	Exploited asymmetric filters and bottleneck layer to lessen the computational cost of deep architectures	Complex architecture design Lack of homogeneity
Highway networks	Introduced training mechanism for deep networks Used auxiliary connections in addition to direct connections	Parametric gating mechanism, difficult to implement
Inception-ResNet	Combined the power of residual learning and inception block	–
Inception-V4	Deep hierarchies of features, multilevel feature representation	Slow in learning
ResNet	Decreased the error rate for deeper networks Introduced the idea of residual learning Alleviates the effect of vanishing gradient problem	A little complex architecture Degrades information of feature-map in feed forwarding Over adaption of hyper-parameters for specific task, due to the stacking of same modules

Table 7 Major challenges associated with implementation of Multi-Path based CNN architectures

Architecture	Strength	Gaps
Multi-path	Shortcut paths provides the option to skip some layers. Different types of the shortcut connections used in literature are zero padded, projection, dropout, 1×1 connections, etc.	
Highway networks	Mitigates the limitations of deep networks by introducing cross layer connectivity	Gates are data dependent and thus may become parameter expensive
ResNet	Use of identity based skip connections to enable cross layer connectivity Information flow gates are data independent and parameter free Can easily pass the signal in both directions, forward and backward	Many layers may contribute very little or no information Relearning of redundant feature-maps may happen
DenseNet	Introduced depth or cross-layer dimension Ensures maximum data flow between the layers in the network Avoid relearning of redundant feature-maps Low and high level both features are accessible to decision layers	Large increase in parameters due to increase in number of feature-maps at each layer

Table 8 Major challenges associated with implementation of width based CNN architectures

Architecture	Strength	Gaps
Width	Earlier, it was assumed that to improve accuracy, the number of layers have to be increased. However, by increasing the number of layers, the vanishing gradient problem arises and training might get slow. So, the concept of widening a layer was also investigated	
Wide ResNet	Showes the effectiveness of parallel use of transformations by increasing the width of ResNet and decreasing its depth Enables feature reuse Have shown that dropouts between the convolutional layer are more effective	Over fitting may occur More parameters than thin deep networks
Pyramidal Net	Introduces the idea of increasing the width gradually per unit Avoids rapid information loss Covers all possible locations instead of maintaining the same dimension till last unit	High spatial and time complexity May become quite complex, if layers are substantially increased
Xception	Introduce the concept that learning across 2D followed by 1D is easier than to learn filters in 3D space Depth-wise separable convolution is introduced Use of cardinality to learn good abstractions	High computational cost
Inception	Varying size filters inside inception module increases the output of the intermediate layers Varying size filters are helpful to capture the diversity in high-detail images	Increase in space and time complexity
ResNeXt	Introduced cardinality to avail diverse transformations at each layer Easy parameter customization due to homogenous topology Uses grouped convolution	High computational cost

Table 9 Major challenges associated with implementation of feature-map exploitation based CNN architectures

Architecture	Strength	Gaps
Feature-map selection	As the deep learning topology is extended, more and more features maps are generated at each step. Many of the Feature-maps might be important for classification task, others might redundant or less important. Hence, feature-map selection is another important dimension in deep learning architectures	
Squeeze and excitation network	It is a block-based concept Introduced a generic block that can be added easily in any CNN model due to its simplicity Squeezes less important features and vice versa	In ResNet, it only considers the residual information for determining the weight of each channel
Competitive Squeeze and excitation networks	Uses feature-map wise statistics from both residual and identity mapping based features Makes a competition between residual and identity feature-maps	Doesn't support the concept of attention

Table 10 Major challenges associated with implementation of channel boosting based CNN architectures

Architecture	Strength	Gaps
Channel boosting	The learning of CNN also relies on the input representation. The lack of diversity and absence of class discernable information in the input may affect CNN performance. For this purpose, the concept of channel boosting (input channel dimension) using auxiliary learners is introduced in CNN to boost the representation of the network (Khan et al. 2018a)	
Channel boosted CNN using transfer learning	It boosts the number of input channels for improving the representational capacity of the network Inductive Transfer Learning is used in a novel way to build a boosted input representation for CNN	Increases in computational load may happen due to the generation of auxiliary channels

Table 11 Major challenges associated with implementation of attention based CNN architectures

Architecture	Strength	Gaps
Attention	Attention networks advantages to choose which patch is the area of the focus or most important in an image	
Residual attention neural network	Generates attention aware feature-maps Easy to scale up due to residual learning Provides different representations of the focused patches Adds soft weights on features using bottom up top-down feedforward attention	Complex model
Convolutional block attention module	CBAM is a generic block designed for feed forward convolutional neural networks Generate both feature-map and spatial attention in a sequential manner Channel attention maps help what to focus Spatial attention helps where to focus Increases efficient flow of information Uses global average pooling and max pool simultaneously	Increase in computational load may happen

to the performance of CNN on different ML tasks have interesting discussions. Some of the challenges faced during the training of deep CNN models are given below:

- Deep CNNs are generally like a black box and thus may lack in interpretation and explanation. Therefore, sometimes it is difficult to verify them.
- Szegedy et al. (2014) showed that training of CNN on noisy image data could cause an increase of misclassification error. The addition of the small quantity of random noise in the input image is capable of fooling the network in such a way that the model will classify the original and its slightly perturbed version differently.
- Each layer of CNN automatically tries to extract better and problem-specific features related to the task. However, for some tasks, it is imperative to know the nature of features extracted by the deep CNNs before classification. The idea of feature visualization in CNNs can help in this direction. Similarly, Hinton reported that lower layers should handover their knowledge only to the relevant neurons of the next layer. In this regard, Hinton proposed an interesting Capsule Network approach (de Vries et al. 2016; Hinton et al. 2018).
- Deep CNNs are based on supervised learning mechanisms, and therefore, the availability of large and annotated data is required for its proper learning. In contrast, humans can learn and generalize from a few examples.
- Hyper-parameter selection highly influences the performance of CNN. A little change in the hyper-parameter values can affect the overall performance of a CNN. That is why careful selection of hyper-parameters is a major design issue that needs to be addressed through some suitable optimization strategy.
- The efficient training of CNN demands powerful hardware resources such as GPUs. However, it is still needed to employ CNNs in embedded and smart devices efficiently. A few applications of deep learning in embedded systems are wound intensity correction, law enforcement in smart cities, etc., (Hinton et al. 2011, 2012a; Lu et al. 2017a).
- In vision-related tasks, one shortcoming of CNN is that it is generally unable to show good performance when used to estimate the pose, orientation, and location of an object. In 2012, AlexNet solved this problem to some extent by introducing the concept of data augmentation. Data augmentation can help CNN in learning diverse internal representations, which ultimately may lead to improved performance.

7 Future directions

The exploitation of different innovative ideas in CNN architectural design has changed the direction of research, especially in image processing and CV. Good performance of CNN on a grid-like topological data presents it as a powerful representational model for images. Architectural design of CNN is a promising research field and in future, it is likely to be one of the most widely used AI techniques.

- Ensemble learning is one of the prospective areas of research in CNNs (Marmanis et al. 2016; Ahmed et al. 2019). The combination of multiple and diverse architectures can aid the model in improving generalization and robustness on diverse categories of images by extracting different levels of semantic representations. Similarly, concepts such as batch normalization, dropout, and new activation functions are also worth mentioning.

- The potential of a CNN as a generative learner is exploited in image segmentation tasks, where it has shown good results (Kahng et al. 2019). The exploitation of generative learning capabilities of CNNs at feature extraction stages can boost the representational power of the model. Similarly, new paradigms are needed that can enhance the learning capacity of CNN by incorporating informative feature-maps that can be learned using auxiliary learners at the intermediate stages of CNN (Khan et al. 2018a).
- In the human visual system, attention is one of the important mechanisms in capturing information from images. The attention mechanism operates in such a way that it not only extracts the essential information from image but also stores its contextual relation with other components of image (Bhunia et al. 2019). In the future, research may be carried out in the direction that preserves the spatial relevance of objects along with their discriminating features at later stages of learning.
- The learning capacity of CNN is generally enhanced by increasing the size of the network, and it can be done in a reasonable time with the help of the current advanced hardware technology such as Nvidia DGX-2 supercomputer. However, the training of deep and high capacity architectures is still a significant overhead on memory usage and computational resources (Lacey et al. 2016; Sze et al. 2017; Justus et al. 2019). Consequently, we still require many improvements in hardware technology that can accelerate research in CNNs. The main concern with CNNs is the run-time applicability. Moreover, the use of CNN is hindered in small hardware, especially in mobile devices, because of its high computational cost. In this regard, different hardware accelerators are needed for reducing both execution time and power consumption (Geng et al. 2019). Some of the very interesting accelerators are already proposed. For example, Application Specific Integrated Circuits, FPGA, and Eyeriss are well known (Moons and Verhelst 2017). Moreover, different operations have been performed to minimize the hardware resources in terms of chip area and energy requirement, by reducing floating-point precision of operands and ternary quantization or minimizing the number of matrix operations. Now it is also time to redirect research towards hardware-oriented approximation models (Geng et al. 2019).
- Deep CNN has a large number of hyper-parameters such as activation function, kernel size, number of neurons per layers, and arrangement of layers, etc. The selection of hyper-parameters and the evaluation time of a deep network, make parameter tuning quite a difficult job. Hyper-parameter tuning is a tedious and intuition driven task, which cannot be defined via explicit formulation. In this regard, Genetic algorithms can also be used to automatically optimize the hyper-parameters by performing searches both in a random fashion as well as by directing search by utilizing previous results (Young et al. 2015; Suganuma et al. 2017; Khan et al. 2019).
- In order to overcome hardware limitations, the concept of pipeline parallelism can be exploited to scale up deep CNN training. Google group has proposed a distributed ML library GPipe (Huang et al. 2018) that offers a model parallelism option for training. In the future, the concept of pipelining can be used to accelerate the training of large models and to scale the performance without tuning hyper-parameters.
- In the future, it is expected that the potential of cloud-based platforms will be exploited for the development of computationally intensive CNN applications (Akar et al. 2019; Stefanini et al. 2019). Deep and wide CNNs present a critical challenge in implementing and executing them on resource-limited devices. Cloud computing not only allows dealing with a massive amount of data but also leverages the benefit of high computational efficiency at a negligible cost. World-leading companies such as Amazon, Microsoft, Google, and IBM offer the public cloud computing facilities at high scalability,

speed, and flexibility to train resource-hungry CNN architectures. Moreover, the cloud environment makes it easy to configure libraries both for researchers and new practitioners.

- CNNs are mostly used for image processing applications, and therefore, the implementation of the state-of-the-art CNN architectures on sequential data requires the conversion of 1D-data into 2D-data. Due to the good feature extraction ability and efficient computations with the limited number of parameters, the trend of using 1D-CNNs is being promoted for sequential data (Vinayakumar et al. 2017; Madrazo et al. 2019).
- Recently high energy physicists at CERN have also been utilizing the learning capability of CNN for the analysis of particle collisions. It is expected that the use of ML and specifically that of deep CNN in high energy physics will grow (Aurisano et al. 2016; Madrazo et al. 2019).

8 Conclusion

CNN has made remarkable progress, especially in image processing and vision-related tasks, and has thus revived the interest of researchers in ANNs. In this context, several research works have been carried out to improve CNN's performance on such tasks. The advancements in CNNs can be categorized in different ways, including activation, loss function, optimization, regularization, learning algorithms, and innovations in architecture. This paper reviews advancement in the CNN architectures, especially based on the design patterns of the processing units and thus has proposed the taxonomy for recent CNN architectures. In addition to the categorization of CNNs into different classes, this paper also covers the history of CNNs, its applications, challenges, and future directions.

The learning capacity of CNN is significantly improved over the years by exploiting depth and other structural modifications. It is observed in recent literature that the main boost in CNN performance has been achieved by replacing the conventional layer structure with blocks. Nowadays, one of the paradigms of research in CNN architectures is the development of new and effective block architectures. The role of a block in a network can be that of an auxiliary learner. These auxiliary learners either exploit spatial or feature-map information or even boost input channels to improve performance. These blocks play a significant role in boosting CNN performance by making problem-aware learning.

Moreover, the block-based architecture of CNN encourages learning in a modular fashion and thereby, making architecture simpler and more understandable. The concept of the block being a structural unit is going to persist and further enhance CNN performance. Additionally, the idea of attention and exploitation of channel information, in addition to spatial information, is expected to gain more importance.

Acknowledgements The authors would like to thank Pattern Recognition lab at DCIS, and PIEAS for providing them computational facilities. The authors express their gratitude to M. Waleed Khan of PIEAS for the detailed discussion related to the Mathematical description of the different CNN architectures.

References

Abbas Q, Ibrahim MEA, Jaffar MA (2019) A comprehensive review of recent advances on deep vision systems. Artif Intell Rev 52:39–76. https://doi.org/10.1007/s10462-018-9633-3

Abdel-Hamid O, Mohamed AR, Jiang H, Penn G (2012) Applying convolutional neural networks concepts to hybrid NN-HMM model for speech recognition. In: ICASSP, IEEE international conference on acoustics speech and signal processing, pp 4277–4280. https://doi.org/10.1007/978-3-319-96145-3_2
Abdel-Hamid O, Deng L, Yu D (2013) Exploring convolutional neural network structures and optimization techniques for speech recognition. In: Interspeech, pp 1173–1175
Abdeljaber O, Avci O, Kiranyaz S et al (2017) Real-time vibration-based structural damage detection using one-dimensional convolutional neural networks. J Sound Vib. https://doi.org/10.1016/j.jsv.2016.10.043
Abdulkader A (2006) Two-tier approach for Arabic offline handwriting recognition. In: Tenth international workshop on frontiers in handwriting recognition
Ahmed U, Khan A, Khan SH et al (2019) Transfer learning and meta classification based deep churn prediction system for telecom industry, pp 1–10
Akar E, Marques O, Andrews WA, Furht B (2019) Cloud-based skin lesion diagnosis system using convolutional neural networks. In: Intelligent computing-proceedings of the computing conference, pp 982–1000
Amer M, Maul T (2019) A review of modularization techniques in artificial neural networks. Artif Intell Rev 52:527–561. https://doi.org/10.1007/s10462-019-09706-7
Aurisano A, Radovic A, Rocco D et al (2016) A convolutional neural network neutrino event classifier. J Instrum. https://doi.org/10.1088/1748-0221/11/09/P09001
Aziz A, Sohail A, Fahad L, et al (2020) Channel Boosted Convolutional Neural Network for Classification of Mitotic Nuclei using Histopathological Images. In: 2020 17th International Bhurban Conference on Applied Sciences and Technology (IBCAST). pp 277–284
Badrinarayanan V, Kendall A, Cipolla R (2017) SegNet: a Deep convolutional encoder-decoder architecture for image segmentation. IEEE Trans Pattern Anal Mach Intell. https://doi.org/10.1109/TPAMI.2016.2644615
Batmaz Z, Yurekli A, Bilge A, Kaleli C (2019) A review on deep learning for recommender systems: challenges and remedies. Artif Intell Rev 52:1–37. https://doi.org/10.1007/s10462-018-9654-y
Bay H, Ess A, Tuytelaars T, Van Gool L (2008) Speeded-up robust features (SURF). Comput Vis Image Underst 110:346–359. https://doi.org/10.1016/j.cviu.2007.09.014
Bengio Y (2009) Learning deep architectures for AI. Found Trends® Mach Learn 2:1–127. https://doi.org/10.1561/2200000006
Bengio Y (2013) Deep learning of representations: looking forward. In: International conference on statistical language and speech processing. Springer, pp 1–37
Bengio Y, Lamblin P, Popovici D, Larochelle H (2007) Greedy layer-wise training of deep networks. In: Advances in neural information processing systems. The MIT Press, pp 153–160
Bengio Y, Courville A, Vincent P (2013) Representation learning: a review and new perspectives. IEEE Trans Pattern Anal Mach Intell 35:1798–1828. https://doi.org/10.1109/TPAMI.2013.50
Berg A, Deng J, Fei-Fei L (2010) Large scale visual recognition challenge 2010
Bettoni M, Urgese G, Kobayashi Y, et al (2017) A convolutional neural network fully implemented on FPGA for embedded platforms. IEEE, pp 49–52. https://doi.org/10.1109/ngcas.2017.16
Bhunia AK, Konwer A, Bhunia AK et al (2019) Script identification in natural scene image and video frames using an attention based Convolutional-LSTM network. Pattern Recognit 85:172–184
Boureau Y (2009) Icml2010B.Pdf. doi: citeulike-article-id:8496352
Bouvrie J (2006) 1 Introduction Notes on Convolutional Neural Networks. doi: http://dx.doi.org/10.1016/j.protcy.2014.09.007
Bulat A, Tzimiropoulos G (2016) Human pose estimation via convolutional part heatmap regression BT. In: Leibe B, Matas J, Sebe N, Welling M (eds) Computer vision—ECCV. Springer, Cham, pp 717–732
Cai Z, Vasconcelos N (2019) Cascade R-CNN: high quality object detection and instance segmentation. IEEE Trans Pattern Anal Mach Intell. https://doi.org/10.1109/tpami.2019.2956516
Chapelle O (1998) Support vector machines for image classification. Stage deuxième année magistère d'informatique l'École Norm Supérieur Lyon 10:1055–1064. https://doi.org/10.1109/72.788646
Chellapilla K, Puri S, Simard P (2006) High performance convolutional neural networks for document processing. In: Tenth international workshop on frontiers in handwriting recognition
Chen Y-N, Han C-C, Wang C-T et al (2006) The application of a convolution neural network on face and license plate detection. In: 18th international conference on pattern recognition, 2006. ICPR 2006, pp 552–555
Chen W, Wilson JT, Tyree S et al (2015) Compressing neural networks with the hashing trick. In: 32nd international conference on machine learning, ICML 2015

Chevalier M, Thome N, Cord M et al (2015) LR-CNN for fine-grained classification with varying resolution. In: 2015 IEEE international conference on image processing (ICIP). IEEE, pp 3101–3105
Chollet F (2017) Xception: deep learning with depthwise separable convolutions. arXiv:1610.02357
Chouhan N, Khan A (2019) Network anomaly detection using channel boosted and residual learning based deep convolutional neural network. Appl Soft Comput 83:105612
Cireşan DC, Meier U, Gambardella LM, Schmidhuber J (2010) Deep, big, simple neural nets for handwritten. Neural Comput 22:3207–3220
Cireşan DC, Meier U, Masci J et al (2011) High-performance neural networks for visual object classification. Preprint arXiv:1102.0183
Cireşan D, Meier U, Masci J, Schmidhuber J (2012a) Multi-column deep neural network for traffic sign classification. Neural Netw 32:333–338. https://doi.org/10.1016/j.neunet.2012.02.023
Cireşan D, Giusti A, Gambardella LM, Schmidhuber J (2012b) Deep neural networks segment neuronal membranes in electron microscopy images. In: Advances in neural information processing systems, pp 2843–2851
Cireşan DC, Giusti A, Gambardella LM, Schmidhuber J (2013) Mitosis detection in breast cancer histology images with deep neural networks BT. In: Proceedings of medical image computing and computer-assisted intervention, MICCAI 2013, pp 411–418
Cireşan DC, Cireşan DC, Meier U, Schmidhuber J (2018) Multi-column deep neural networks for image classification. In: IEEE conference on computer vision and pattern recognition
Collobert R, Weston J (2008) A unified architecture for natural language processing: Deep neural networks with multitask learning. In: Proceedings of the 25th international conference on Machine learning. ACM, pp 160–167
Csáji B (2001) Approximation with artificial neural networks. M.Sc. Thesis 45
Dahl G, Mohamed A, Hinton GE (2010) Phone recognition with the mean-covariance restricted Boltzmann machine. In: Advances in neural information processing systems, pp 469–477
Dahl GE, Sainath TN, Hinton GE (2013) Improving deep neural networks for LVCSR using rectified linear units and dropout. In: 2013 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE, pp 8609–8613
Dai J, Li Y, He K, Sun J (2016) R-FCN: object detection via region-based fully convolutional networks. J Power Sources. https://doi.org/10.1016/j.jpowsour.2007.02.075
Dalal N, Triggs W (2004) Histograms of oriented gradients for human detection. In: IEEE computer society conference on computer vision and pattern recognition CVPR05, vol. 1, pp 886–893. https://doi.org/10.1109/cvpr.2005.177
Dauphin YN, De Vries H, Bengio Y (2015) Equilibrated adaptive learning rates for non-convex optimization. In: Advances in neural information processing system 2015, January, pp 1504–1512
Dauphin YN, Fan A, Auli M, Grangier D (2017) Language modeling with gated convolutional networks. In: Proceedings of the 34th international conference on machine learning, vol 70, pp 933–941
de Vries H, Memisevic R, Courville A (2016) Deep learning vector quantization. In: European symposium on artificial neural networks, computational intelligence and machine learning
Decoste D, Schölkopf B (2002) Training invariant support vector machines. Mach Learn 46:161–190
Delalleau O, Bengio Y (2011) Shallow versus deep sum-product networks. In: Advances in neural information processing systems, pp 666–674
Deng L (2012) The MNIST database of handwritten digit images for machine learning research [best of the web]. IEEE Signal Process Mag 29:141–142
Deng L, Yu D, Delft B (2013) Deep learning: methods and applications foundations and trends R in signal processing. Sig Process 7:3–4. https://doi.org/10.1561/2000000039
Do MN, Vetterli M (2005) The contourlet transform: an efficient directional multiresolution image representation. IEEE Trans Image Process 14:2091–2106
Dollár P, Tu Z, Perona P, Belongie S (2009) Integral channel features
Donahue J, Anne Hendricks L, Guadarrama S et al (2015) Long-term recurrent convolutional networks for visual recognition and description. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 2625–2634
Dong C, Loy CC, He K, Tang X (2016) Image super-resolution using deep convolutional networks. IEEE Trans Pattern Anal Mach Intell 38:295–307
Erhan D, Bengio Y, Courville A, Vincent P (2009) Visualizing higher-layer features of a deep network. Univ Montr 1341:1
Farfade SS, Saberian MJ, Li L-J (2015) Multi-view face detection using deep convolutional neural networks. In: Proceedings of the 5th ACM on international conference on multimedia retrieval—ICMR'15. ACM Press, New York, USA, pp 643–650

Fasel B (2002) Facial expression analysis using shape and motion information extracted by convolutional neural networks. In: Proceedings of the 2002 12th IEEE workshop on neural networks for signal processing, 2002, pp 607–616

Frizzi S, Kaabi R, Bouchouicha M et al (2016) Convolutional neural network for video fire and smoke detection. In: IECON 2016-42nd annual conference of the IEEE industrial electronics society. IEEE, pp 877–882

Frome A, Cheung G, Abdulkader A, et al (2009) Large-scale privacy protection in Google Street View. In: Proceedings of the IEEE international conference on computer vision

Frosst N, Hinton G (2018) Distilling a neural network into a soft decision tree. In: CEUR workshop proceedings

Fukushima K (1988) Neocognitron: a hierarchical neural network capable of visual pattern recognition. Neural Netw 1:119–130

Fukushima K, Miyake S (1982) Neocognitron: a self-organizing neural network model for a mechanism of visual pattern recognition. In: Competition and cooperation in neural nets. Springer, pp 267–285

Garcia C, Delakis M (2004) Convolutional face finder: a neural architecture for fast and robust face detection. IEEE Trans Pattern Anal Mach Intell. https://doi.org/10.1109/TPAMI.2004.97

Gardner MW, Dorling SR (1998) Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences. Atmos Environ 32:2627–2636

Geng X, Lin J, Zhao B et al (2019) Hardware-aware softmax approximation for deep neural networks. In: Lecture notes in computer science. Lecture notes in artificial intelligence, Lecture notes in bioinformatics. pp 107–122

Gidaris S, Komodakis N (2015) Object detection via a multi-region and semantic segmentation-aware U model. In: Proceedings of IEEE international conference on computer vision 2015, pp 1134–1142. https://doi.org/10.1109/iccv.2015.135

Girshick R (2015) Fast R-CNN. In: Proceedings of the IEEE international conference on computer vision

Giusti A, Cireşan DC, Masci J et al (2013) Fast image scanning with deep max-pooling convolutional neural networks. In: 2013 IEEE international conference on image processing. IEEE, pp 4034–4038

Glorot X, Bengio Y (2010) Understanding the difficulty of training deep feedforward neural networks. In: Proceedings of the thirteenth international conference on artificial intelligence and statistics, pp 249–256

Goh H, Thome N, Cord M, Lim J-H (2013) Top-down regularization of deep belief networks. In: Advances in neural information processing systems (NIPS). pp 1878–1886

Goodfellow I, Bengio Y, Courville A (2017) Deep learning. Nat Methods 13:35. https://doi.org/10.1038/nmeth.3707

Grill-Spector K, Weiner KS, Gomez J et al (2018) The functional neuroanatomy of face perception: from brain measurements to deep neural networks. Interface Focus 8:20180013. https://doi.org/10.1098/rsfs.2018.0013

Grün F, Rupprecht C, Navab N, Tombari F (2016) A taxonomy and library for visualizing learned features in convolutional neural networks. https://doi.org/10.1080/10962247.2014.948229

Gu J, Wang Z, Kuen J et al (2018) Recent advances in convolutional neural networks. Pattern Recognit 77:354–377. https://doi.org/10.1016/j.patcog.2017.10.013

Guo Y, Liu Y, Oerlemans A et al (2016) Deep learning for visual understanding: a review. Neurocomputing 187:27–48. https://doi.org/10.1016/j.neucom.2015.09.116

Hamel P, Eck D (2010) Learning features from music audio with deep belief networks. In: ISMIR, Utrecht, The Netherlands, pp 339–344

Han S, Mao H, Dally WJ (2016) Deep compression: compressing deep neural networks with pruning, trained quantization and Huffman coding. In: 4th international conference on learning representations, ICLR 2016—conference track proceedings

Han D, Kim J, Kim J (2017) Deep pyramidal residual networks. In: 2017 IEEE conference on computer vision and pattern recognition (CVPR). IEEE, pp 6307–6315

Han W, Feng R, Wang L, Gao L (2018) Adaptive spatial-scale-aware deep convolutional neural network for high-resolution remote sensing imagery scene classification. In: IGARSS 2018–2018 IEEE international geoscience and remote sensing symposium, pp 4736–4739. https://doi.org/10.1109/igarss.2018.8518290

Hanin B, Sellke M (2017) Approximating continuous functions by ReLU Nets of minimal width. Preprint. arXiv:1710.11278

He K, Zhang X, Ren S, Sun J (2015a) Deep residual learning for image recognition. Multimed Tools Appl 77:10437–10453. https://doi.org/10.1007/s11042-017-4440-4

He K, Zhang X, Ren S, Sun J (2015b) Spatial pyramid pooling in deep convolutional networks for visual recognition. IEEE Trans Pattern Anal Mach Intell 37:1904–1916
He K, Gkioxari G, Dollar P, Girshick R (2017) Mask R-CNN. In: Proceedings of the IEEE international conference on computer vision
Heikkilä M, Pietikäinen M, Schmid C (2009) Description of interest regions with local binary patterns. Pattern Recognit 42:425–436. https://doi.org/10.1016/j.patcog.2008.08.014
Hinton GE, Osindero S, Teh Y-W (2006) A fast learning algorithm for deep belief nets. Neural Comput 18:1527–1554
Hinton GE, Krizhevsky A, Wang SD (2011) Transforming auto-encoders. In: International conference on artificial neural networks. Springer, pp 44–51
Hinton G, Deng L, Yu D et al (2012a) Deep neural networks for acoustic modeling in speech recognition: the shared views of four research groups. IEEE Signal Process Mag 29:82–97
Hinton GE, Srivastava N, Krizhevsky A, et al (2012b) Improving neural networks by preventing co-adaptation of feature detectors. pp 1–18. arXiv:12070580
Hinton G, Sabour S, Frosst N (2018) Matrix capsules with EM routing. In: 6th international conference on learning representations, ICLR 2018 - conference track proceedings
Hochreiter S (1998) The vanishing gradient problem during learning recurrent neural nets and problem solutions. Int J Uncertain Fuzziness Knowl-Based Syst 6:107–116
Howard AG, Zhu M, Chen B, et al (2017) MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. arXiv:170404861
Hu B, Lu Z, Li H, Chen Q (2011) Topic modeling for named entity queries. In: Proceedings of the 20th ACM international conference on Information and knowledge management—CIKM'11. ACM Press, New York, New York, USA, 2009
Hu J, Shen L, Sun G (2018a) Squeeze-and-excitation networks. In: 2018 IEEE/CVF conference on computer vision and pattern recognition. IEEE, pp 7132–7141
Hu Y, Wen G, Luo M, et al (2018b) Competitive inner-imaging squeeze and excitation for residual network. arXiv:1807.08920v3
Huang G, Sun Y, Liu Z et al (2016a) Deep networks with stochastic depth. In: European conference on computer vision. Springer, pp 646–661
Huang G, Sun Y, Liu Z et al (2016b) Deep networks with stochastic depth BT. In: European conference on computer vision ECCV 2016. Springer, pp 646–661
Huang G, Liu Z, Van Der Maaten L, Weinberger KQ (2017) Densely connected convolutional networks. In: Proceedings of 30th IEEE conference on computer vision and pattern recognition, CVPR 2017, pp 2261–2269. https://doi.org/10.1109/cvpr.2017.243
Huang Y, Cheng Y, Chen D et al (2018) GPipe: efficient training of giant neural networks using pipeline parallelism. arXiv:1811.06965v3
Huang KY, Wu CH, Hong QB et al (2019) Speech emotion recognition using deep neural network considering verbal and nonverbal speech sounds. In: Proceedings of IEEE international conference on acoustics, speech and signal processing ICASSP
Hubel DH, Wiesel TN (1959) Receptive fields of single neurones in the cat's striate cortex. J Physiol. https://doi.org/10.1113/jphysiol.1959.sp006308
Hubel DH, Wiesel TN (1962) Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. J Physiol 160:106–154. https://doi.org/10.1113/jphysiol.1962.sp006837
Hubel DH, Wiesel TN (1968) Receptive fields and functional architecture of monkey striate cortex. J Physiol 195:215–243. https://doi.org/10.1113/jphysiol.1968.sp008455
Ioffe S, Szegedy C (2015) Batch normalization: accelerating deep network training by reducing internal covariate shift. J Mol Struct. https://doi.org/10.1016/j.molstruc.2016.12.061
Jaderberg M, Simonyan K, Zisserman A, Kavukcuoglu K (2015) Spatial transformer networks. Nature. https://doi.org/10.1038/nbt.3343
Jarrett K, Kavukcuoglu K, Ranzato M, LeCun Y (2009) What is the best multi-stage architecture for object recognition? In: IEEE 12th international conference on comput vision, 2009, pp 2146–2153
Ji S, Yang M, Yu K, Xu W (2010) 3D convolutional neural networks for human action recognition. Int Conf Mach Learn 35:221–231. https://doi.org/10.1109/TPAMI.2012.59
Joachims T (1998) Text categorization with support vector machines: Learning with many relevant features. In: European conference on machine learning. pp 137–142
Justus D, Brennan J, Bonner S, McGough AS (2019) Predicting the computational cost of deep learning models. In: Proceedings of 2018 IEEE international conference on big data, Big Data 2018
Kafi M, Maleki M, Davoodian N (2015) Functional histology of the ovarian follicles as determined by follicular fluid concentrations of steroids and IGF-1 in Camelus dromedarius. Res Vet Sci 99:37–40. https://doi.org/10.1016/j.rvsc.2015.01.001

Kahng M, Thorat N, Chau DHP et al (2019) GAN Lab: understanding complex deep generative models using interactive visual experimentation. IEEE Trans Vis Comput Graph 25:310–320
Kalchbrenner N, Grefenstette E, Blunsom P (2014) A convolutional neural network for modelling sentences. Preprint arXiv:1404.2188
Kawashima T, Kawanishi Y, Ide I et al (2017) Action recognition from extremely low-resolution thermal image sequence. In: 2017 14th IEEE international conference on advanced video and signal based surveillance, AVSS 2017. IEEE, pp 1–6
Kawaguchi K, Huang J, Kaelbling LP (2019) Effect of depth and width on local minima in deep learning. Neural Comput 31:1462–1498. https://doi.org/10.1162/neco_a_01195
Khan A, Sohail A, Ali A (2018a) A New channel boosted convolutional neural network using transfer learning. Preprint arXiv:1804.08528
Khan A, Zameer A, Jamal T, Raza A (2018b) Deep belief networks based feature generation and regression for predicting wind power. Preprint arXiv:1807.11682
Khan A, Qureshi AS, Hussain M et al (2019) A recent survey on the applications of genetic programming in image processing. Preprint arXiv:1901.07387
Krizhevsky A, Sutskever I, Hinton GE (2012) ImageNet classification with deep convolutional neural networks. Adv Neural Inf Process Syst. https://doi.org/10.1061/(ASCE)GT.1943-5606.0001284
Kuen J, Kong X, Wang G et al (2017) DelugeNets: deep networks with efficient and flexible cross-layer information inflows. In: 2017 IEEE international conference on computer vision workshop (ICCVW), pp 958–966
Kuen J, Kong X, Wang G, Tan YP (2018) DelugeNets: deep networks with efficient and flexible cross-layer information inflows. In: Proceedings of IEEE international conference on computer vision work ICCVW 2017, pp 958–966. https://doi.org/10.1109/iccvw.2017.117
Lacey G, Taylor GW, Areibi S (2016) Deep learning on FPGAs: past, present, and future. arXiv:1602042 83
Larsson G, Maire M, Shakhnarovich G (2016) Fractalnet: ultra-deep neural networks without residuals. Preprint 1605.07648, pp 1–11
Laskar MNU, Giraldo LGS, Schwartz O (2018) Correspondence of deep neural networks and the brain for visual textures, pp 1–17
Le QV, Ranzato M, Monga R et al (2011) Building high-level features using large scale unsupervised learning. In: IEEE International conference on acoustics speech and signal processing ICASSP, pp 8595–8598. https://doi.org/10.1109/icassp.2013.6639343
LeCun Y (2007) Effcient BackPrp. J Exp Psychol Gen 136:23–42
LeCun Y, Boser B, Denker JS et al (1989) Backpropagation applied to handwritten zip code recognition. Neural Comput 1:541–551
LeCun Y, Jackel LD, Bottou L et al (1995) Learning algorithms for classification: a comparison on handwritten digit recognition. Neural Netw Stat Mech Perspect 261:276
LeCun Y, Bottou L, Bengio Y, Haffner P (1998) Gradient-based learning applied to document recognition. Proc IEEE 86:2278–2324
LeCun Y, Kavukcuoglu K, Farabet CC et al (2010) Convolutional networks and applications in vision. In: ISCAS. IEEE, pp 253–256
LeCun Y, Bengio Y, Hinton G (2015) Deep learning. Nature 521:436–444. https://doi.org/10.1038/nature14539
Lee C-Y, Gallagher PW, Tu Z (2016) Generalizing pooling functions in convolutional neural networks: mixed, gated, and tree. In: Artificial intelligence and statistics, pp 464–472
Lee S, Son K, Kim H, Park J (2017) Car plate recognition based on CNN using embedded system with GPU, pp 239–241
Levi G, Hassner T (2009) Sicherheit und Medien. Sicherheit und Medien. https://doi.org/10.1109/CVPRW.2015.7301352
Li S, Liu Z-Q, Chan AB (2014) Heterogeneous multi-task learning for human pose estimation with deep convolutional neural network. In: 2014 IEEE conference on computer vision and pattern recognition workshops. IEEE, pp 488–495
Li H, Lin Z, Shen X et al (2015) A convolutional neural network cascade for face detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 5325–5334
Li X, Bing L, Lam W, Shi B (2018) Transformation networks for target-oriented sentiment classification, pp 946–956
Lin M, Chen Q, Yan S (2013) Network in network, pp 1–10. https://doi.org/10.1109/asru.2015.7404828
Lin T-Y, Maire M, Belongie S et al (2014) Microsoft coco: common objects in context. In: European conference on computer vision. Springer, pp 740–755

Lin TY, Dollár P, Girshick R et al (2017) Feature pyramid networks for object detection. In: Proceedings of 30th IEEE conference on computer vision and pattern recognition, CVPR 2017
Lindholm E, Nickolls J, Oberman S, Montrym J (2008) NVIDIA TESLA: a unified graphics and computing architecture. IEEE Micro 28:39–55. https://doi.org/10.1109/MM.2008.31
Linnainmaa S (1970) The representation of the cumulative rounding error of an algorithm as a Taylor expansion of the local rounding errors. Master's Thesis (in Finnish), Univ Helsinki 6–7
Liu C-L, Nakashima K, Sako H, Fujisawa H (2003) Handwritten digit recognition: benchmarking of state-of-the-art techniques. Pattern Recognit 36:2271–2285
Liu W, Wang Z, Liu X et al (2017) A survey of deep neural network architectures and their applications. Neurocomputing 234:11–26. https://doi.org/10.1016/j.neucom.2016.12.038
Liu X, Deng Z, Yang Y (2019) Recent progress in semantic image segmentation. Artif Intell Rev 52:1089–1106. https://doi.org/10.1007/s10462-018-9641-3
Long ZM, Guo SQ, Chen GJ, Yin BL (2012) Modeling and simulation for the articulated robotic arm test system of the combination drive. In: 2011 international conference on mechatronics and materials engineering ICMME 2011, pp 151:480–483. https://doi.org/10.4028/www.scientific.net/AMM.151.480
Long J, Shelhamer E, Darrell T (2015) Fully convolutional networks for semantic segmentation. In: 2015 IEEE conference on computer vision and pattern recognition (CVPR). IEEE, pp 3431–3440
Lowe DG (1999) Object recognition from local scale-invariant features. In: Proceedings of Seventh IEEE International Conference on Computer Vision, vol 2, pp 1150–1157. https://doi.org/10.1109/iccv.1999.790410
Lowe DG (2004) Distinctive image features from scale-invariant keypoints. Int J Comput Vis 60:91–110
Lu H, Li B, Zhu J et al (2017a) Wound intensity correction and segmentation with convolutional neural networks. Concurr Comput Pract Exp 29:e3927
Lu Z, Pu H, Wang F et al (2017b) The expressive power of neural networks: a view from the width. In: Advances in neural information processing systems, pp 6231–6239
Lv E, Wang X, Cheng Y, Yu Q (2019) Deep ensemble network based on multi-path fusion. Artif Intell Rev 52:151–168. https://doi.org/10.1007/s10462-019-09708-5
Madrazo CF, Heredia I, Lloret L, Marco de Lucas J (2019) Application of a convolutional neural network for image classification for the analysis of collisions in high energy physics. EPJ Web Conf. https://doi.org/10.1051/epjconf/201921406017
Mao X, Shen C, Yang Y-B (2016) Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections. In: Advances in neural information processing systems, pp 2802–2810
Marmanis D, Wegner JD, Galliani S et al (2016) Semantic segmentation of aerial images with an ensemble of CNNs. ISPRS Ann Photogramm Remote Sens Spat Inf Sci 3:473
Matsugu M, Mori K, Ishii M, Mitarai Y (2002) Convolutional spiking neural network model for robust face detection. In: Proceedings of the 9th international conference on neural information processing, 2002. ICONIP'02, pp 660–664
Mikolov T, Karafiát M, Burget L et al (2010) Recurrent neural network based language model. In: Eleventh annual conference of the international speech communication association
Misra D (2019) Mish: a self regularized non-monotonic neural activation function. arXiv:190808681
Mohamed A, Dahl GE, Hinton G (2012) Acoustic modeling using deep belief networks. IEEE Trans Audio Speech Lang Process 20:14–22
Montufar GF, Pascanu R, Cho K, Bengio Y (2014) On the number of linear regions of deep neural networks. In: Advances in neural information processing systems, pp 2924–2932
Moons B, Verhelst M (2017) An energy-efficient precision-scalable ConvNet processor in 40-nm CMOS. IEEE J Solid-State Circuits 52:903–914
Morar A, Moldoveanu F, Gröller E (2012) Image segmentation based on active contours without edges. In: IEEE 8th international conference on intelligent computer communication processing ICCP 2012, pp 213–220. https://doi.org/10.1109/iccp.2012.6356188
Nair V, Hinton GE (2010) Rectified linear units improve restricted Boltzmann machines. In: ICML 27th international conference on machine learning
Najafabadi MM, Villanustre F, Khoshgoftaar TM et al (2015) Deep learning applications and challenges in big data analytics. J Big Data 2:1–21. https://doi.org/10.1186/s40537-014-0007-7
Nguyen Q, Mukkamala M, Hein M (2018) Neural networks should be wide enough to learn disconnected decision regions. Preprint arXiv:1803.00094
Nguyen G, Dlugolinsky S, Bobák M et al (2019) Machine learning and deep learning frameworks and libraries for large-scale data mining: a survey. Artif Intell Rev 52:77–124. https://doi.org/10.1007/s10462-018-09679-z

Nickolls J, Buck I, Garland M, Skadron K (2008) Scalable parallel programming with CUDA. In: ACM SIGGRAPH 2008 classes on SIGGRAPH'08. ACM Press, New York, New York, USA, p 1

Nwankpa C, Ijomah W, Gachagan A, Marshall S (2018) Activation functions: comparison of trends in practice and research for deep learning. Preprint arXiv:1811.03378

Oh K-S, Jung K (2004) GPU implementation of neural networks. Pattern Recognit 37:1311–1314

Ojala T, Pietikäinen M, Harwood D (1996) A comparative study of texture measures with classification based on feature distributions. Pattern Recognit 29:51–59. https://doi.org/10.1016/0031-3203(95)00067-4

Ojala T, PeitiKainen M, Maenpã T (2002) Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. IEEE Trans Pattern Anal Mach Intell 247:971–987

Oquab M, Bottou L, Laptev I, Sivic J (2014) Learning and transferring mid-level image representations using convolutional neural networks. In: Proceedings of the IEEE computer society conference on computer vision and pattern recognition. IEEE, pp 1717–1724

Pang J, Chen K, Shi J et al (2020) Libra R-CNN: towards balanced learning for object detection

Pascanu R, Mikolov T, Bengio Y (2012) Understanding the exploding gradient problem. arXiv :1211.5063

Peng X, Hoffman J, Yu SX, Saenko K (2016) Fine-to-coarse knowledge transfer for low-res image classification. In: 2016 IEEE international conference on image processing (ICIP). IEEE, pp 3683–3687

Potluri S, Fasih A, Vutukuru LK et al (2011) CNN based high performance computing for real time image processing on GPU. In: Proceedings of the joint INDS'11 & ISTET'11, pp 1–7

Qureshi AS, Khan A (2018) Adaptive transfer learning in deep neural networks: wind power prediction using knowledge transfer from region to region and between different task domains. Preprint arXiv :1810.12611

Qureshi AS, Khan A, Zameer A, Usman A (2017) Wind power prediction using deep neural network based meta regression and transfer learning. Appl Soft Comput J 58:742–755. https://doi.org/10.1016/j.asoc.2017.05.031

Ramachandran P, Zoph B, Le QV (2017) Swish: a self-gated activation function

Ranjan R, Patel VM, Chellappa R (2015) A deep pyramid deformable part model for face detection. Preprint arXiv:1508.04389

Ranzato M, Huang FJ, Boureau YL, LeCun Y (2007) Unsupervised learning of invariant feature hierarchies with applications to object recognition. In: Proceedings of the IEEE computer society conference on computer vision and pattern recognition. IEEE, pp 1–8

Rawat W, Wang Z (2016) Deep convolutional neural networks for image classification: a comprehensive review. Neural Comput 61:1120–1132. https://doi.org/10.1162/NECO

Ren S, He K, Girshick R, Sun J (2015) Faster R-CNN: towards real-time object detection with region proposal networks. Adv Neural Inf Process Syst. https://doi.org/10.1109/tpami.2016.2577031

Ronneberger O, Fischer P, Brox T (2015) U-net: convolutional networks for biomedical image segmentation. In: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)

Roy AG, Navab N, Wachinger C (2018) Concurrent spatial and channel 'squeeze & excitation' in fully convolutional networks. Lecture Notes in Computer Science (including Subser Lectue Notes in Artificial Intelligence Lecture Notes in Bioinformatics) 11070 LNCS:421–429. https://doi.org/10.1007/978-3-030-00928-1_48

Russakovsky O, Deng J, Su H et al (2015) imagenet large scale visual recognition challenge. Int J Comput Vis. https://doi.org/10.1007/s11263-015-0816-y

Salakhutdinov R, Larochelle H (2010) Efficient learning of deep Boltzmann machines. In: Proceedings of the thirteenth international conference on artificial intelligence and statistics, pp 693–700

Scherer D, Müller A, Behnke S (2010) Evaluation of pooling operations in convolutional architectures for object recognition. In: Artificial neural networks–ICANN 2010. Springer, pp 92–101

Schmidhuber J (2007) New millennium AI and the convergence of history. In: Challenges for computational intelligence. Springer, pp 15–35

Sermanet P, Chintala S, Lecun Y (2012) Convolutional neural networks applied to house numbers digit classification. In: Proceedings of the 21st international conference on pattern recognition (ICPR2012), Tsukuba. IEEE, pp 3288–3291

Shakeel MF, Bajwa NA, Anwaar AM et al (2019) Detecting driver drowsiness in real time through deep learning based object detection. In: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)

Sharma A, Muttoo SK (2018) Spatial image steganalysis based on ResNeXt. In: 2018 IEEE 18th International conference on communication technology, pp 1213–1216. https://doi.org/10.1109/icct.2018.8600132

Shi Y, Tian Y, Wang Y, Huang T (2017) Sequential deep trajectory descriptor for action recognition with three-stream CNN. IEEE Trans Multimed 19:1510–1520
Shin H-CC, Roth HR, Gao M et al (2016) Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning. IEEE Trans Med Imaging 35:1285–1298. https://doi.org/10.1109/TMI.2016.2528162
Simard PY, Steinkraus D, Platt JC (2003) Best practices for convolutional neural networks applied to visual document analysis, p 958
Simonyan K, Zisserman A (2014) Two-stream convolutional networks for action recognition in videos. In: Advances in neural information processing systems, pp 568–576
Simonyan K, Zisserman A (2015) Very deep convolutional networks for large-scale image recognition. ICLR 75:398–406. https://doi.org/10.2146/ajhp170251
Simonyan K, Vedaldi A, Zisserman A (2013) Deep inside convolutional networks: visualising image classification models and saliency maps, pp 1–8. https://doi.org/10.1080/00994480.2000.10748487
Sinha T, Verma B, Haidar A (2018) Optimization of convolutional neural network parameters for image classification. In: 2017 IEEE symposium series on computational intelligence SSCI 2017, pp 1–7. https://doi.org/10.1109/ssci.2017.8285338
Spanhol FA, Oliveira LS, Petitjean C, Heutte L (2016a) A dataset for breast cancer histopathological image classification. IEEE Trans Biomed Eng 63:1455–1462
Spanhol FA, Oliveira LS, Petitjean C, Heutte L (2016b) Breast cancer histopathological image classification using convolutional neural networks. In: 2016 international joint conference on neural networks (IJCNN). IEEE, pp 2560–2567
Srinivas S, Sarvadevabhatla RK, Mopuri KR et al (2016) A taxonomy of deep convolutional neural nets for computer vision. Front Robot AI 2:1–13. https://doi.org/10.3389/frobt.2015.00036
Srivastava N, Hinton G, Krizhevsky A et al (2014) Dropout: a simple way to prevent neural networks from overfittin. J Mach Learn Res 1:11. https://doi.org/10.1016/j.micromeso.2003.09.025
Srivastava RK, Greff K, Schmidhuber J (2015a) Highway networks. https://doi.org/10.1002/esp.3417
Srivastava RK, Greff K, Schmidhuber J (2015b) Training very deep networks. In: Advances in neural information processing systems
Stefanini M, Lancellotti R, Baraldi L, Calderara S (2019) A deep-learning-based approach to vm behavior identification in cloud systems. In: Proceedings of the 9th international conference on cloud computing and services science. SCITEPRESS—Science and Technology Publications, pp 308–315
Strigl D, Kofler K, Podlipnig S (2010) Performance and scalability of GPU-based convolutional neural networks. In: 2010 18th Euromicro international conference on parallel, distributed and network-based processing (PDP), pp 317–324
Suganuma M, Shirakawa S, Nagao T (2017) A genetic programming approach to designing convolutional neural network architectures. In: Proceedings of the genetic and evolutionary computation conference. ACM, pp 497–504
Sun L, Jia K, Yeung D-Y, Shi BE (2015) Human action recognition using factorized spatio-temporal convolutional networks. In: Proceedings of the IEEE international conference on computer vision, pp 4597–4605
Sundermeyer M, Schlüter R, Ney H (2012) LSTM neural networks for language modeling. In: Thirteenth annual conference of the international speech communication association
Sze V, Chen YH, Yang TJ, Emer JS (2017) Efficient processing of deep neural networks: a tutorial and survey. In: Proceedings of IEEE
Szegedy C, Zaremba W, Sutskever I et al (2014) Intriguing properties of neural networks. In: 2nd international conference on learning Representations, ICLR 2014 - conference track proceedings
Szegedy C, Liu W, Jia Y et al (2015) Going deeper with convolutions. In: 2015 IEEE conference on computer vision and pattern recognition (CVPR). IEEE, pp 1–9
Szegedy C, Ioffe S, Vanhoucke V (2016a) Inception-v4, Inception-ResNet and the impact of residual connections on learning. Preprint arXiv:1602.07261v2 131:262–263. https://doi.org/10.1007/s10236-015-0809-y
Szegedy C, Vanhoucke V, Ioffe S et al (2016b) Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE Computer Society conference on computer vision and pattern recognition. IEEE, pp 2818–2826
Targ S, Almeida D, Lyman K (2016) Resnet in Resnet: generalizing residual architectures. Preprint arXiv:1603.08029
Tong W, Song L, Yang X, et al (2015) CNN-based shot boundary detection and video annotation. In: 2015 IEEE international symposium on broadband multimedia systems and broadcasting. IEEE, pp 1–5
Tong T, Li G, Liu X, Gao Q (2017) Image super-resolution using dense skip connections. In: 2017 IEEE international conference on computer vision (ICCV), pp 4809–4817
Tran D, Bourdev L, Fergus R, et al (2015) Learning spatiotemporal features with 3D convolutional networks. In: Proceedings of the IEEE international conference on computer vision, pp 4489–4497

Ullah A, Ahmad J, Muhammad K et al (2017) Action recognition in video sequences using deep bi-directional LSTM with CNN features. IEEE Access 6:1155–1166

Vinayakumar R, Soman KP, Poornachandrany P (2017) Applying convolutional neural network for network intrusion detection. In: 2017 International conference on advances in computing, communications and informatics, ICACCI 2017

Vincent P, Larochelle H, Bengio Y, Manzagol P-A (2008) Extracting and composing robust features with denoising autoencoders. In: Proceedings of the 25th international conference on machine learning. ACM, pp 1096–1103

Vinyals O, Toshev A, Bengio S, Erhan D (2017) Show and tell: lessons learned from the 2015 MSCOCO image captioning challenge. IEEE Trans Pattern Anal Mach Intell. https://doi.org/10.1109/TPAMI.2016.2587640

Wahab N, Khan A, Lee YS (2017) Two-phase deep convolutional neural network for reducing class skewness in histopathological images based breast cancer detection. Comput Biol Med 85:86–97. https://doi.org/10.1016/j.compbiomed.2017.04.012

Wahab N, Khan A, Lee YS (2019) Transfer learning based deep CNN for segmentation and detection of mitoses in breast cancer histopathological images. Microscopy 68:216–233. https://doi.org/10.1093/jmicro/dfz002

Wang H, Raj B (2017) On the origin of deep learning, pp 1–72. https://doi.org/10.1016/0014-5793(91)81229-2

Wang H, Schmid C (2013) Action recognition with improved trajectories. In: Proceedings of the IEEE international conference on computer vision, pp 3551–3558

Wang T, Wu DJDJ, Coates A, Ng AY (2012) End-to-end text recognition with convolutional neural networks. In: International Conference on Pattern Recognition ICPR, pp 3304–3308

Wang F, Jiang M, Qian C et al (2017a) Residual attention network for image classification. In: 2017 IEEE conference on computer vision and pattern recognition (CVPR). IEEE, pp 6450–6458

Wang X, Gao L, Song J, Shen H (2017b) Beyond frame-level CNN: saliency-aware 3-D CNN With LSTM for video action recognition. IEEE Signal Process Lett 24:510–514. https://doi.org/10.1109/LSP.2016.2611485

Wang Y, Wang L, Wang H, Li P (2019) End-to-end image super-resolution via deep and shallow convolutional networks. IEEE Access 7:31959–31970. https://doi.org/10.1109/ACCESS.2019.2903582

Woo S, Park J, Lee JY, Kweon IS (2018) CBAM: Convolutional block attention module. Lect Notes Comput Sci (including Subser Lect Notes Artif Intell Lect Notes Bioinformatics) 11211 LNCS:3–19. https://doi.org/10.1007/978-3-030-01234-2_1

Wu J, Leng C, Wang Y, et al (2016) Quantized convolutional neural networks for mobile devices. In: Proceedings of the IEEE computer society conference on computer vision and pattern recognition

Xie S, Girshick R, Dollar P et al (2017) Aggregated residual transformations for deep neural networks. In: 2017 IEEE conference on computer vision and pattern recognition (CVPR). IEEE, pp 5987–5995

Xie W, Zhang C, Zhang Y et al (2018) An energy-efficient FPGA-based embedded system for CNN application. In: 2018 IEEE international conference on electron devices and solid state circuits (EDSSC). IEEE, pp 1–2

Xiong Y, Kim HJ, Hedau V (2019) ANTNets: mobile convolutional neural networks for resource efficient image classification. arXiv:190403775

Xu B, Wang N, Chen T, Li M (2015a) Empirical evaluation of rectified activations in convolutional network. J Foot Ankle Res 1:O22. https://doi.org/10.1186/1757-1146-1-S1-O22

Xu K, Ba J, Kiros R et al (2015b) Show, attend and tell: neural image caption generation with visual attention. In: International conference on machine learning, pp 2048–2057

Yamada Y, Iwamura M, Kise K (2016) Deep pyramidal residual networks with separated stochastic depth. Preprint arXiv:1612.01230

Yang Q, Pan SJ, Yang Q, Fellow QY (2008) A survey on transfer learning. IEEE Trans Knowl Data Eng 1:1–15. https://doi.org/10.1109/TKDE.2009.191

Yang S, Luo P, Loy C-C, Tang X (2015) From facial parts responses to face detection: a deep learning approach. In: Proceedings of the IEEE international conference on computer visio, pp 3676–3684

Yang J, Xiong W, Li S, Xu C (2019) Learning structured and non-redundant representations with deep neural networks. Pattern Recognit 86:224–235

Yıldırım Ö, Pławiak P, Tan RS, Acharya UR (2018) Arrhythmia detection using deep convolutional neural network with long duration ECG signals. Comput Biol Med. https://doi.org/10.1016/j.compbiomed.2018.09.009

Young SR, Rose DC, Karnowski TP et al (2015) Optimizing deep learning hyper-parameters through an evolutionary algorithm. In: Proceedings of the workshop on machine learning in high-performance computing environments. ACM, p 4

Zagoruyko S, Komodakis N (2016) Wide residual networks. Proc Br Mach Vis Conf 87(1-87):12. https://doi.org/10.5244/C.30.87

Zeiler MD, Fergus R (2013) Visualizing and understanding convolutional networks. Preprint arXiv:1311.2901v3, vol 30, pp 225–231. https://doi.org/10.1111/j.1475-4932.1954.tb03086.x

Zhang X, LeCun Y (2015) Text understanding from scratch. Preprint arXiv:1502.01710

Zhang K, Zhang Z, Li Z et al (2016) Joint face detection and alignment using multitask cascaded convolutional networks. IEEE Signal Process Lett 23:1499–1503

Zhang X, Li Z, Loy CC, Lin D (2017) PolyNet: a pursuit of structural diversity in very deep networks. In: Proceedings of 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, pp 3900–3908. https://doi.org/10.1109/cvpr.2017.415

Zhang X, Zhou X, Lin M, Sun J (2018a) ShuffleNet: an extremely efficient convolutional neural network for mobile devices. In: Proceedings of the IEEE computer society conference on computer vision and pattern recognition

Zhang Y, Qiu Z, Yao T, et al (2018b) Fully convolutional adaptation networks for semantic segmentation. In: Proceedings of the IEEE computer society conference on computer vision and pattern recognition

Zhang Q, Zhang M, Chen T et al (2019) Recent advances in convolutional neural network acceleration. Neurocomputing 323:37–51. https://doi.org/10.1016/j.neucom.2018.09.038

Zheng H, Fu J, Mei T, Luo J (2017) Learning multi-attention convolutional neural network for fine-grained image recognition. In: 2017 IEEE international conference on computer vision (ICCV), pp 5219–5227

Zhou B, Khosla A, Lapedriza A et al (2016) Learning deep features for discriminative localization. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 2921–2929