A Survey of the Recent Architectures of Deep Convolutional Neural Networks

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

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

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

Abstract

Deep Convolutional Neural Networks (CNNs) are a special type of Neural Networks, which have shown state-of-the-art results on various competitive benchmarks. The powerful learning ability of deep CNN is largely achieved with the use of multiple feature extraction stages that can automatically learn hierarchical representations from the data. Availability of a large amount of data and improvements in the hardware processing units have accelerated the research in CNNs, and recently very interesting deep CNN architectures are reported. The recent race in developing deep CNN architectures has shown that the innovative architectural ideas, as well as parameter optimization, can improve the CNN performance on various vision-related tasks. In this regard, different ideas in the CNN design have been explored such as the use of different activation and loss functions, parameter optimization, regularization, and restructuring of the processing units. However, the major improvement in representational capacity of the deep CNN is achieved by the restructuring of the processing units. Especially, the idea of using a block as a structural unit instead of a layer is receiving substantial attention. 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, it covers the elementary understanding of the CNN components and sheds light on the current challenges and applications of CNNs.

Keywords: Deep Learning, Classification, Convolutional Neural Networks, Representational Capacity, Residual Learning, Channel Boosted CNN, VGG, AlexNet, Inception, and ResNet.

1. Introduction

Machine Learning (ML) algorithms belong to a specialized area of research in Artificial Intelligence (AI), which endows intelligence to computers by learning the underlying relationships among the data and making decisions without being explicitly programmed. Different ML algorithms have been developed since the late 1990s, for the emulation of human sensory responses such as speech and vision, but they have failed to achieve human-level satisfaction [1–6]. The challenging nature of Machine Vision (MV) gives rise to a specialized class of Neural Networks (NN) [7], which mimics processing of Visual Cortex and is called Convolutional Neural Network (CNN).

The architectural design of CNN was inspired by Hubel and Wiesel's work (1962) [8] and was based on primate's visual cortex structure. CNN first came to limelight through the work of LeCuN in 1989 for the processing of grid-like topological data (images and time series data) [9,10]. CNNs are considered as one of the best techniques for understanding image content and have shown state-of-the-art results on image recognition, segmentation, detection, and retrieval related tasks [11]. The success of CNN 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 [12]. At present, most of the frontrunners of image processing competitions are employing deep CNN based models.

Topology of CNN is divided into multiple learning stages composed of a combination of the convolutional layer, non-linear processing units, and subsampling layers [13]. Each layer performs multiple transformations using a bank of convolutional kernels (filters) [7]. Convolution operation extracts locally correlated features by dividing the image into small slices (similar to the retina of the human eye), making it capable of learning interesting features. Output of the convolutional kernels is assigned to non-linear processing units, which not only helps in learning abstraction 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. Output of the non-linear function is usually followed by subsampling, which also makes the input invariant to geometrical distortions [7,14].

CNN got famous due to its hierarchical feature extraction ability. Hierarchical organization of CNN emulates the deep and layered based learning process of the primary sensorial areas of the Neocortex in the human brain, which automatically extract features from the underlying data [15]. CNN has the ability to extract low, mid, and high-level features. Higher features (more abstract features) are a combination of lower and mid-level features. With automatic feature extraction ability, CNN reduces the need for synthesizing a separate feature extractor. Thus, CNN can learn good internal representation from raw pixels with diminutive processing.

Deep-layered hierarchical representation of CNN mimics primate's ventral pathway of visual cortex (V1-V2-V4-IT/VTC) [16]. The visual cortex of primates first receives input from the retinotopic area, where multi-scale highpass filtering and contrast normalization is performed by the lateral geniculate nucleus. After this, detection is performed by different regions of the visual cortex categorized as V1, V2, V3, and V4. In fact, V1 and V2 portion of visual cortex are similar to convolutional, and subsampling layers, whereas inferior temporal region resembles the higher layers of CNN, which makes inference about the image [17]. During training, CNN learns through backpropagation algorithm, by regulating the change in weights with respect to the input. Optimization of a relevant cost function by CNN using backpropagation algorithm is similar to the response based learning of human brain.

Deep architecture has an advantage over shallow architectures when dealing with complex learning problems. Stacking of multiple linear and non-linear processing units in a layer wise fashion gives deep networks the ability to learn complex representations at different levels of abstraction. The main boom in the use of CNN for image classification and segmentation occurred after it was observed that the representational capacity of CNN can be enhanced with an increased depth [18]. In addition, the availability of high computing resources is also one of the main reasons of the popularity of deep CNNs. Apart from supervised learning, deep CNNs have potential to learn useful representation from large amounts of unlabeled data. Different level of features including both low and high-level CNN features can be transferred to a generic recognition task by exploiting the concept of Transfer Learning (TL) [19,20]. Use of the multiple mapping functions by CNN enables it to improve the learning and extraction of invariant representations and consequently, makes it capable to handle recognition tasks of hundreds or thousands of categories. Key advantages of CNN are hierarchical learning, automatic feature extraction, multi-tasking, and weight sharing [21,22]. In addition to this, the major motivation associated with CNN architectures is that learning process (feature extraction) can be visualized in a layer wise fashion.

Different innovations in CNN architecture have been proposed since 2012. These innovations can be categorized as parameter optimization, regularization, structural reformulation, etc. However, it is observed that the main improvement in CNN performance was mainly due to restructuring of processing units and designing of new blocks. CNN based applications became prevalent after the marvelous performance of AlexNet on ImageNet dataset [18]. Similarly, Zeiler and Fergus [23] introduced the concept of layer-wise visualization of features, which shifted the trend towards extraction of features at low spatial resolution with deep architecture such as VGG [24]. Nowadays, most of the new architectures are built upon the principle of simple and homogenous topology introduced by VGG. On the other hand, Google group introduced a very famous idea of split, transform and merge, 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 [25]. In 2015, the concept of skip connections introduced by ResNet [26] for the training of deep CNNs got famous, and latterly, this concept was used by most of the succeeding Nets, such as Inception-ResNet [27], WideResNet [28], ResNext [29] etc. In surge of improving the learning capacity of CNN, different architectural designs such as WideResNet, Pyramidal Nets, Xception etc. explored the effect of multilevel transformations in terms of an additional cardinality and increase in width [28–30]. Therefore, the focus of research shifted from parameter optimization and connections readjustment towards improved architectural design (layer structure) of the network. This resulted in many new architectural ideas such as channel boosting, spatial and channel wise exploitation and attention based information processing etc. in CNN. [31–33].

Different surveys are conducted on deep CNNs in the past few years that elaborate the basic components of CNN and their alternatives. The survey presented in [34] has reviewed the famous architectures from 2012-2015 along with their components. Similarly, the survey presented in [35] discussed taxonomy of CNNs based on acceleration techniques. Similarly, in the literature, there are surveys that focus on different applications of CNN [22,36]. However, this survey discusses the intrinsic taxonomy present in the recent and prominent CNN architectures. All of the CNN architectures discussed in this survey can be broadly classified into seven main categories. The paper is organized in the following order (shown in Figure 1). Section 1 summarizes the underlying basis of CNN, their resemblance with primate's visual cortex, and their contribution in MV. In this regard, Section 2 provides the overview on basic CNN components and Section 3 discusses the architectural evolution of deep CNNs. Whereas, Section 4, discusses the recent innovations in CNN architectures and categorizes CNNs into seven broad classes. Section 5 and 6 shed light on applications of CNNs and current challenges, whereas the last section draws conclusion and discusses future work.

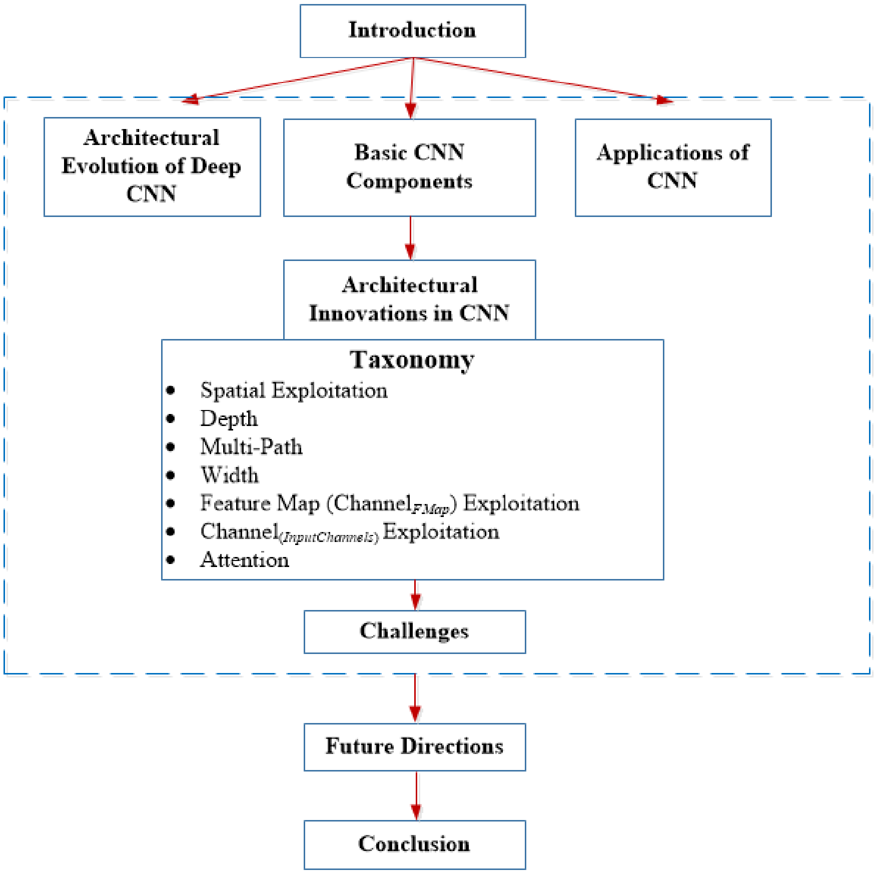
Introduction
Architectural Evolution of Deep CNN
Basic CNN Components
Applications of CNN
Architectural Innovations in CNN
Taxonomy
Spatial Exploitation
Depth
Multi-Path
Width
Feature Map (Channel$_{FMap}$) Exploitation
Channel$_{(InputChannels)}$ Exploitation
Attention
Challenges
Future Directions
Conclusion

Figure 1: Organization of the survey paper.

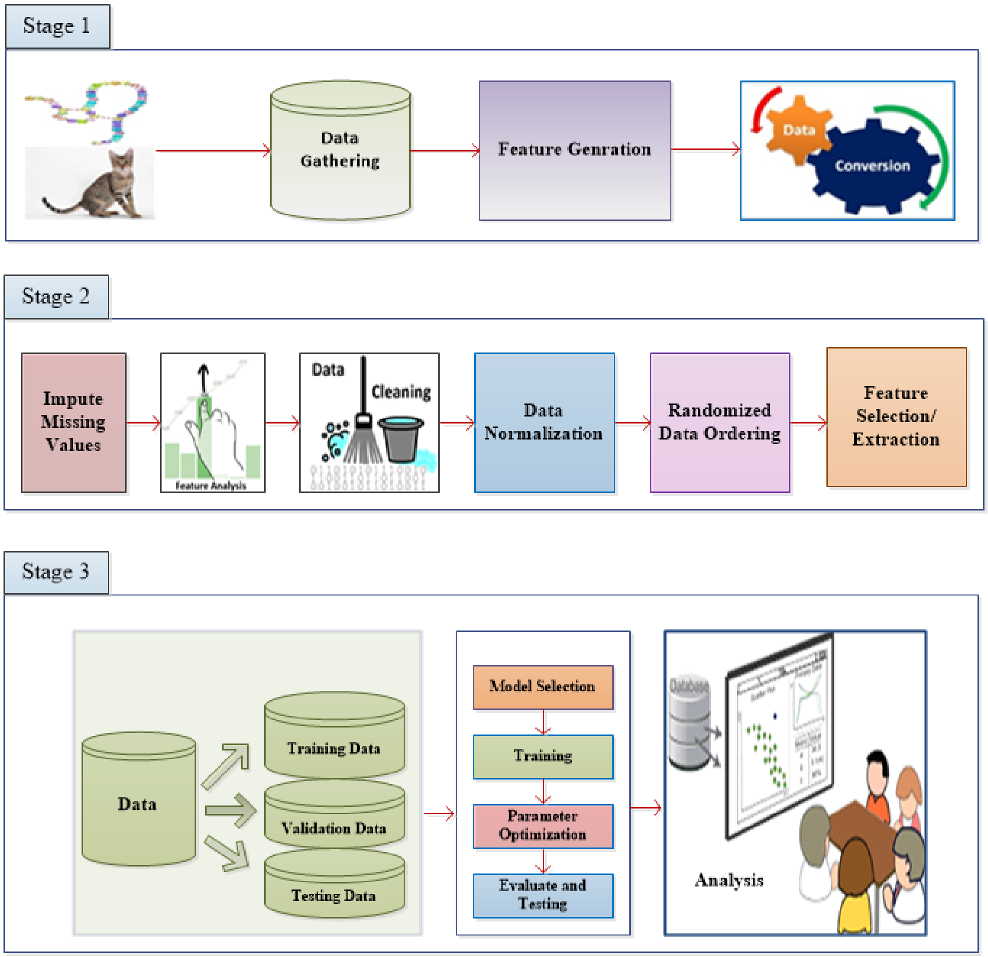

Figure 2: Basic layout of a typical ML system. In ML related tasks, initially data is preprocessed and then assigned to a classification system. A typical ML problem follows three steps: stage 1 is related to data mining, stage 2 performs preprocessing and feature selection, whereas stage 3 is based on model selection, parameter tuning, and analysis. CNN has a good feature extraction and strong discrimination ability, therefore in a ML system; it can be used for data generation, feature extraction and classification.

2. Basic CNN Components

Nowadays, CNN is considered as the most widely used technique of ML; especially in vision related applications and has shown state-of-the-art results in tasks related to ML. A typical block diagram of a ML system is shown in Figure 2. Since, CNN possesses both good feature extraction and strong discrimination ability, therefore in a ML system; it is largely used at feature extraction/generation and model selection stages.

A typical CNN architecture generally comprises of alternate layers of convolution and pooling followed by one or more fully connected layers at the end. In some cases, fully connected layer is replaced with global average pooling layer. In addition to different learning stages, different regulatory units such as batch normalization and dropout are also incorporated to optimize CNN performance [37]. The arrangement of CNN components play a prominent role in designing of new architectures for achieving enhanced performance. This section briefly discusses the role of these components in CNN architecture.

2.1. Convolutional Layer

Convolutional layer is composed of a set of convolutional kernels (each neuron act as a kernel). These kernels are associated with a small area of the image known as a receptive field. It works by dividing the image into small blocks (known as receptive field) and convolving them with a specific set of weights (multiplying elements of the filter (weights) with corresponding receptive field elements) [37]. Convolution operation is explained in equation (1).

$$F_l^k = (I_{x,y} * K_l^k) \qquad (1)$$

Input image is represented by $I_{x,y}$, x, y shows spatial locality where K_l^k represents l^{th} convolutional kernel of the k^{th} layer. Division of image into small blocks helps in extracting locally correlated pixel values. This locally aggregated information is also known as feature motifs. Different set of features are extracted by sliding convolutional kernel on the image with the same set of weights. 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 [38].

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 downsampling like convolution, 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 [39].

$$Z_l = f_p(F_{x,y}^l) \qquad (2)$$

Equation (2) shows the pooling operation in which Z_l represents l^{th} output feature map, $F^{l}_{x,y}$ shows l^{th} input feature map, where f_p *(.)* defines 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 [14,40]. Furthermore, pooling may also help in increasing the generalization by reducing overfitting. In addition to this, reduction in the size of feature maps regulates complexity of the network. Different types of pooling formulations such as max, average, L2, and overlapping pooling are used for extracting translational invariant features [41,42].

2.3. Activation Function

Activation function serves as a decision function and helps in learning a complex pattern. Selection of an appropriate activation function can accelerate the learning process. Activation function for convolved feature map is defined in equation (3).

$$T_l^k = f_A(F_l^k) \tag{3}$$

In above equation (3), F_l^k is an output of a convolution operation, which is assigned to activation function; $f_A(.)$ that adds non-linearity and returns a transformed output T_l^k for k^{th} layer. In literature, therefore, different activation functions such as sigmoid, tanh, maxout, ReLU, and variants of ReLU such as leaky ReLU, ELU, and PReLU [34,41,43,44] are used to inculcate non-linear combination of features. ReLU and its variants are preferred over others activations as it helps in overcoming the vanishing gradient problem [45].

2.4. Batch Normalization

Batch normalization is used to address the issues related to internal covariance shift within feature maps. The internal covariance shift is a change in the distribution of hidden units' values, which slow down the convergence (by forcing learning rate to small value) and requires careful initialization of parameters. Batch normalization for transformed feature map T_l^k is shown in equation (4).

$$N_l^k = \frac{T_l^k}{\sigma^2 + \sum_i T_i^k} \qquad (4)$$

In equation (4), N_l^k represents normalized feature map, where T_l^k is input feature map and σ depicts variation in feature map. Batch normalization unifies the distribution of feature map values by bringing them to zero mean and unit variance [46]. Furthermore, it smoothen the flow of gradient and acts as a regulating factor, which improves generalization of the network without relying on dropout.

2.5. Dropout

Dropout introduces regularization within 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. This random dropping of some connections or units produces several thinned network architectures out of which one representative network is selected with small weights. This selected architecture is then considered as an approximation of all of the proposed networks [47].

2.6. Fully Connected Layer

Fully connected layer is mostly used at the end of the network for classification purpose. It takes input from the previous layer and analyses output of all previous layers globally [48]. It makes a non-linear combination of selected features, which are used for the classification of data. Unlike pooling and convolution, it is a global operation [49].

3. Architectural Evolution of Deep CNN

CNNs are the most famous algorithms among biologically inspired AI techniques. CNN history began from neurobiological experiments conducted by Hubel and Wiesel (1959, 1962) [8]. Their work provided a platform for many cognitive models, all of which were latterly replaced by CNN. Over the decades, different efforts have been imposed to improve the performance of CNN. These improvements can be categorized into five different eras and are discussed below.

3.1. Late 1980s-1999: Origin of CNN

CNNs have been applied to visual tasks since the late 1980s. In 1989, LeCuN et al. proposed the first multilayered CNN named as ConvNet, whose origin was rooted in Fukushima's Neocognitron [50,51]. LeCuN proposed supervised training of ConvNet, using Backpropagation algorithm [10,52] in comparison to the unsupervised reinforcement learning scheme used by its predecessor Neocognitron, which lead a foundation for the first modern 2d CNN. Supervised training in CNN endows the automatic feature learning ability from raw input rather than designing of handcrafted features, which were required by traditional ML methods. This ConvNet showed successful results for handwritten digit and zip code recognition related problems [53]. In 1998, ConvNet was improved by LeCuN and used for classifying characters in a document recognition application [54]. This modified architecture was named as LeNet-5 [55,56], which was an improvement over initial CNN as it can extract feature representation in a hierarchical way from raw pixels. Reliance of LeNet-5 on fewer parameters along with consideration of spatial topology of images enabled CNN to recognize rotational variants of the image [56]. Due to its good performance in optical character recognition, its commercial use in ATM and Banks was started in 1993 and 1996, respectively. Though, many successful features were gained by LeNet-5, yet the main concern associated with it was that its discrimination power was not scaled to classification tasks other than hand recognition.

3.2. Early 2000: Stagnation of CNN

In the late 1990s and early 2000s, interest in NNs was low and little research was carried out to explore the role of CNNs in different applications such as object detection, video surveillance, etc. Use of CNN in ML tasks became dormant due to insignificant improvement in performance with no noticeable decrease in error. At that time, other statistical methods and, in particular, SVM [57,58] became more popular than CNN due to its good performance [59]. It was widely presumed in early 2000 that the backpropagation algorithm used for training of CNN was not effective in converging to optimal points and learned no useful features in supervised fashion as compared to handcrafted features [60]. Meanwhile, different researchers kept working on CNN and tried to optimize its performance. In 2003, Simard et al. [61] improved CNN architecture and showed good results as compared to SVM on hand digit benchmark dataset, MNIST [6254,59,63].

This performance improvement expedited the research in CNN by extending its application in optical character recognition (OCR) to other script's character recognition [63–65], deployment in image sensors for face detection in video conferencing, and regulation of street crimes, etc. Likewise, CNN based systems were industrialized in markets for tracking and detection of customers [66]. Moreover, CNN's potential in other applications such as medical image segmentation, anomaly detection, and robot vision was also evaluated [67–69].

3.3. 2006-2011: Revival of CNN

Deep NNs have a very complex structure and time intensive training phase that sometimes spanned over weeks and even months. In early 2000, there was no appropriate approach for the training of deep networks. Because of these limitations, CNN was not scaled well enough to be applied to complex problems. These challenges halted the use of CNN in ML related tasks.

To address these problems, in 2006 many methods were developed to overcome the difficulties encountered in the training of deep CNNs and learning of invariant features. Hinton proposed greedy layer-wise training approach in 2006 [70], for deep architectures. The renaissance of a deep learning [71,72] was one of the factor, which brought deep CNNs into the limelight. Huang et al. used max pooling [40] instead of subsampling, which showed good results by learning of invariant features [73]. In late 2006, researchers started using graphics processing units (GPUs) [74] to accelerate training of deep NN and CNN architectures [75,76]. In 2007, NVIDIA launched the CUDA programming platform [77,78], which allows exploitation of parallel processing capabilities of GPU with a much greater degree [78]. The use of GPUs for NN training [75,79] and hardware improvements were the main factor, which revived the research in CNN. In 2010, Fei-Fei Li's group at Stanford, established a large database of images known as ImageNet, containing millions of labeled images [80]. This database was coupled with the annual ILSVRC competition, where submitted models performances were evaluated and scored [81]. This was a turning point in improving the performance and increasing the use of CNN.

3.4. 2012-2014: Rise of CNN

Availability of big training data, hardware advancements, and computational resources contributed to advancement in CNN algorithms and renaissance of CNN in object detection, image classification, and segmentation related tasks [11,82]. However, the success of CNN in image classification tasks was not just the result of aforementioned factors but mainly contributed by architectural modifications, parameter optimization, incorporation of regulatory units, and reformulation and readjustment of connections within the network [34,35,83].

The main breakthrough in CNN performance was brought by AlexNet [18]. AlexNet won the 2012-ILSVRC competition, which was one of the most difficult challenges for image detection and classification. AlexNet improved performance by exploiting depth (incorporate multiple levels of transformation) and introduced regularization term in CNN. The exemplary performance of AlexNet [18] as compared to conventional ML techniques in 2012-ILSVRC (AlexNet reduced error rate from 25.8 to 16.4) suggested that the main reason of the saturation in CNN performance before 2006 was largely due to the unavailability of enough training data and computational resources. These resource deficiencies made it hard to train a high-capacity CNN without deterioration of performance [84].

With CNN becoming more of a commodity in the computer vision (CV) field, a number of attempts have been made to improve the original architecture of AlexNet [18]. Similarly, in 2013 and 2014 researchers worked on parameter optimization to accelerate CNN performance in diverse applications with a small increase in computational complexity. In 2013, Zeiler and Fergus [23] defined a mechanism to visualize learned filters of each CNN layer. Visualization approach was used to improve the feature extraction approach by reducing the size of filters. Similarly, VGG architecture [24] proposed by the Oxford group, which was runner-up at the 2014-ILSVRC competition, made the receptive field much small in comparison to that of AlexNet but with increased volume. In VGG, depth is increased from 9 layers to 16, by making the volume of features maps double at each layer. In the same year, GoogleNet [85] that won 2014-ILSVRC competition, not only exerted its efforts to reduce computational cost by changing layer design, but also widen the width in compliance with depth to improve CNN performance. GoogleNet introduced the concept of split, transform and merge blocks, within which multiscale and multilevel transformation is incorporated to capture both local and global information [27,85,86]. The use of multilevel transformations help CNN in tackling details of images at various levels. In the year 2012-14, the main improvement in the learning capacity of CNN was achieved by increasing its depth and parameter optimization strategies. This suggests that the representation depth is beneficial in improving the generalization of classifier.

3.5. 2015-Present: Rapid increase in Architectural Innovations and Applications of CNN

The major improvements in CNN performance were occurred from 2015-2018. The research in CNN is still on going and it has a significant potential of improvement. Representational depth improves generalization by defining diverse level of features ranging from simple to complex. Multiple levels of transformation make learning easy by chopping complex problems into smaller modules. However, the main challenge faced by deep architectures is the problem of negative learning, which occurs due to diminishing gradient at lower layers of the network. To handle this problem, different research groups worked on readjustment of layers connections and designing of new modules. In earlier 2015, Srivastava et al. [87] used the concept of cross-channel connectivity and information gating mechanism [88,89] to solve the vanishing gradient problem and to improve the network representational capacity. This idea got famous in late 2015 and a similar concept of residual blocks or skip connections was coined [26]. Residual blocks are a variant of cross-channel connectivity, which smoothen learning by regularizing the flow of information across blocks [90–92]. This idea was used by ResNet architecture (proposed by Microsoft) [26] for the training of 150 layers deep network. The idea of cross-channel connectivity is further extended to multilayer connectivity by different researchers to improve representation [93,94] (resulting in Deluge, DenseNet etc).

In the year 2016, the width of the network was also explored in connection with depth to improve feature learning stage [28,30]. Apart from this, no new architectural modification became prominent but instead, different researchers used hybrid of the already proposed architectures to improve deep CNN performance [27,90–92,95]. This fact gave the intuition that there might be other factors more important as compared to the appropriate assembly of the network units that can effectively regulate CNN performance. In this regard, Hu et al. (2017) [96] identified that the network representation has a role in learning of deep CNNs. Hu et al. introduced the idea of feature map exploitation and pinpointed that less informative and domain extraneous features affect the performance of the network to a larger extent. Hu et al. [96] exploited the aforementioned idea and proposed new architecture named as Squeeze and Excitation Network (SE-Network). It exploits feature map (commonly known as channel in literature) information by designing a specialized SE-block. This block assigns weight to each feature map depending upon its contribution in class discrimination. This idea was further developed by different researchers, which assign attention to important regions by exploiting both spatial and channel information [32,33,97]. In 2018, a new idea of channel boosting was introduced by Khan et al [31]. The motivation behind the training of network with boosted channel representation was to use an enriched representation for learning of discriminating features.

From 2012 up till now, a lot of improvements have been reported in CNN architecture. Now the focus of research is towards designing of new blocks that can boost network representation by adding artificial channels or by exploiting both feature maps and spatial information.

4. Architectural Innovations in CNN

Different improvements in CNN architecture have been made from 1989 to date. These improvements can be categorized as parameters optimizations, regularization, structural reformulation, etc. However, it is observed that the main upgradation in CNN performance was mainly due to the restructuring of processing units and designing of new blocks. All of the innovations in CNN have been made in relation with depth and spatial exploitation. Depending upon the type of architectural modification, CNN can be broadly categorized into seven different classes' namely spatial exploitation, depth, multi-path, width, channel boosting, feature map exploitation, and attention based CNNs. Taxonomy of deep CNN architectures is shown in Figure 3. Based on aforementioned taxonomy, CNN architectures are placed into seven different classes and their summary is represented in Table 1.

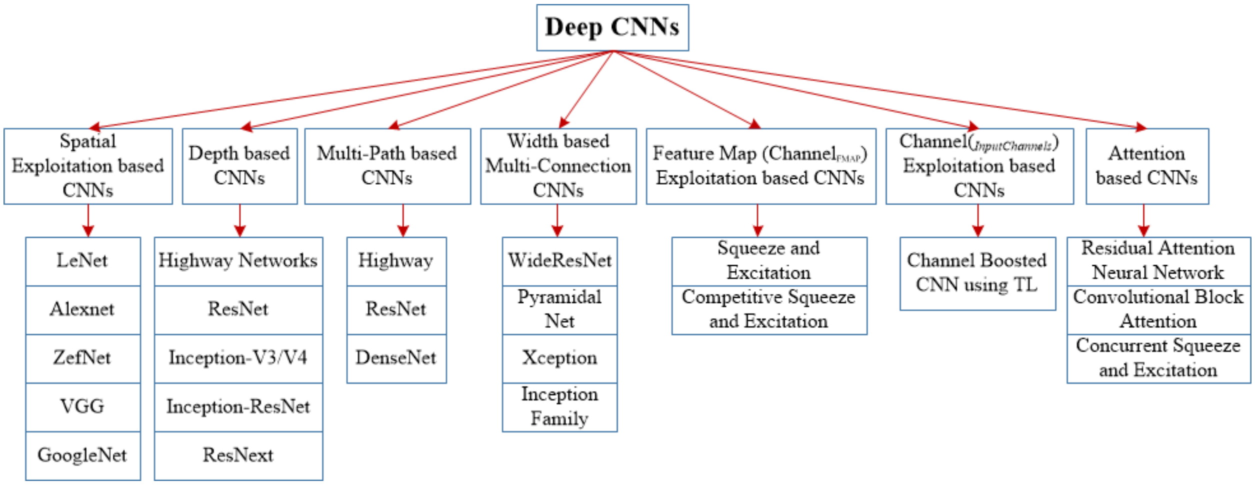
Deep CNNs
Spatial Exploitation based CNNs
Depth based CNNs
Multi-Path based CNNs
Width based Multi-Connection CNNs
Feature Map (Channel$_{FMAP}$) Exploitation based CNNs
Channel($_{InputChannels}$) Exploitation based CNNs
Attention based CNNs
LeNet
Alexnet
ZefNet
VGG
GoogleNet
Highway Networks
ResNet
Inception-V3/V4
Inception-ResNet
ResNext
Highway
ResNet
DenseNet
WideResNet
Pyramidal Net
Xception
Inception Family
Squeeze and Excitation
Competitive Squeeze and Excitation
Channel Boosted CNN using TL
Residual Attention Neural Network
Convolutional Block Attention
Concurrent Squeeze and Excitation

Figure 3: Taxonomy of deep CNN architectures.

4.1. Spatial Exploitation based CNNs

CNN has a large number of parameters and hyperparameters such as the number of processing units (neurons), layers, filter size, stride, learning rate, and activation function, etc. [98,99]. As convolutional operation considers the neighborhood (locality) of input pixels, therefore different levels of correlation can be explored by using a different size of filters. Therefore, in early 2000, researchers exploited spatial transformations to improve performance in this regard; different sizes of filters were explored to evaluate their impact on learning of network. Different size of filters encapsulate different levels of granularity; usually, small size filters extract fine-grained and large size extract coarse-grained information. In this way, by the adjustment of filter sizes, CNN can perform well both on coarse and fine-grained level details.

4.1.1. LeNet

LeNet was proposed by LeCuN in 1998 [56]. 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 [100]. The main limitation of traditional multilayer NN was that it considers each pixel as a separate input and applies a transformation on it, which was a huge computational burden, specifically at that time [101]. LeNet exploited the underlying basis of image that the neighboring pixels are correlated to each other and 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. This changed the conventional view of training where each pixel was considered as a separate input feature from its neighborhood and ignores the correlation among them. LeNet was the first CNN architecture, which not only reduced the number of parameters and computation but ingeniously learned features.

4.1.2. AlexNet

LeNet [56] though begin the history of deep CNNs but at that time CNN was limited to hand digit recognition tasks and not scaled to all classes of images. AlexNet [18] is considered as the first deep CNN architecture, which showed groundbreaking results for image classification and recognition tasks. AlexNet was proposed by Krizhevesky et al. [18] who enhanced the learning capacity of the CNN by making it deeper and by applying a number of parameter optimizations strategies. Basic architectural design of AlexNet is shown in Figure 4. Hardware limitations curtail the learning capacity of deep CNN architecture by restricting to small size. In order to get benefit from capacity of CNN, Alexnet was trained in parallel on two NVIDIA GTX 580 GPUs to overcome shortcomings of hardware. In AlexNet, feature extraction stages were extended from 5 (LeNet) to 7 to make CNN applicable for diverse categories of images. Despite the fact that depth improves generalization for different resolutions of images but the main drawback associated with depth is overfitting. To address this challenge, Krizhevesky et al. exploited the idea of Hinton [47,102], whereby their algorithm randomly skips some transformational units during training to enforce the model to learn features that are more robust. In addition to this, ReLU [43] was employed as a non-saturating activation function to improve the convergence rate by alleviating the problem of vanishing gradient [45]. 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 (11x11 and 5x5) at the initial layers, compared to previously proposed networks. Due to its efficient learning approach, AlexNet has significant importance in the new generation of CNNs and it has started a new era of research in CNNs.

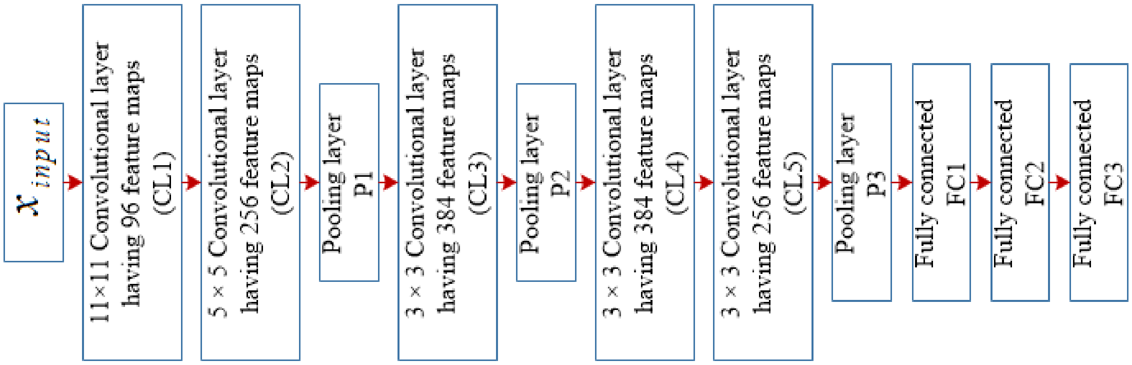
x_{input}
11×11 Convolutional layer having 96 feature maps (CL1)
5 × 5 Convolutional layer having 256 feature maps (CL2)
Pooling layer P1
3 × 3 Convolutional layer having 384 feature maps (CL3)
Pooling layer P2
3 × 3 Convolutional layer having 384 feature maps (CL4)
3 × 3 Convolutional layer having 256 feature maps (CL5)
Pooling layer P3
Fully connected FC1
Fully connected FC2
Fully connected FC3

Figure 4: Basic layout of AlexNet architecture.

4.1.3. ZefNet

Learning mechanism of CNN was largely based on hit-and-trial, without knowing the exact reason behind the improvement before 2013. This lack of understanding limited the performance of deep CNNs on complex images. In 2013, Zeiler and Fergus [23] proposed a multilayer Deconvolutional NN (DeconvNet), which got famous as ZefNet. ZefNet was developed to quantitatively visualize network performance. The idea of visualization of network activity was to monitor CNN performance by interpreting neurons activation. In one of the previous study, Erhan et al. exploited the same idea and optimized performance of Deep Belief Networks (DBNs) by visualizing hidden layers' feature [103]. In the same manner, Le et al. evaluated the performance of deep unsupervised autoencoder (AE) by visualizing the image classes generated by output neurons [104]. 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 convolutional layer back to visually perceptible image patterns consequently gives the neuron level interpretation of the internal feature representation learned at each layer [105,106]. The objective of ZefNet was to monitor the learning scheme during training and used the findings in diagnosing a potential problem associated with the model. This idea was experimentally validated on AlexNet using DeconvNet, which showed that only a few neurons were active, while other neurons were dead (inactive) in the first and second layer of network. Moreover, it showed that the features extracted by second layer exhibit 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 size of filters and stride to retain the maximum number of features in the first two convolutional layers. This readjustment in CNN topology resulted in improvement in performance, which suggested that features visualization can be used for identification of design shortcomings and for timely adjustment of parameters.

4.1.4. VGG

With the successful use of CNNs for image recognition, Simonyan and Zisserman proposed a simple and effective design principle for CNNs. This new architecture was termed as VGG and was modular in layers pattern [24]. VGG was made 19 layers deep than AlexNet [18] and ZefNet [23] to simulate the relation of depth with representational capacity of network. ZefNet, 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 11x11 and 5x5 filters with a stack of 3x3 filters layer and experimentally demonstrated that concurrent placement of 3x3 filters can induce the effect of the large size filter. VGG suggested that parallel placement of small size filters make the receptive field as effective as that of large size filters (5x5 and 7x7). Use of small size filters give an additional benefit of low computational complexity by reducing the number of parameters. These findings set a new trend in research to work with a smaller size filters in CNN. VGG regulate the complexity of network by placing 1x1 convolution in between the convolutional layers, which in addition learn a linear combination of the resultant feature maps. For the tuning of the network, max pooling [40] is placed after convolutional layer and padding was performed to maintain the spatial resolution. VGG showed good results both for image classification and localization problems. Although, VGG was not at top place of 2014-ILSVRC competition but got fame due to its simplicity, homogenous topology, and increased depth. The main limitation associated with VGG was high computational cost. Even with the use of small size filters, VGG suffered from high computational burden due to the use of about 140 million parameters.

4.1.4. GoogleNet

GoogleNet was the winner of the 2014-ILSVRC competition and is also known as Inception-V1. The main objective of the GoogleNet [85] architecture was to achieve high accuracy with a reduced computational cost. It introduced the new concept of inception module (block) in CNN, whereby it incorporates multi-scale convolutional transformations using split, transform and merge idea for feature extraction. Architecture of inception block is shown in Figure 5. This block encapsulates filters of different sizes (1x1, 3x3, and 5x5) to capture spatial information in combination with channel information at different spatial resolutions. In GoogleNet, conventional convolutional layer is replaced by small blocks similar to idea of substituting each layer with micro NN as proposed by Network in Network (NIN) architecture [48]. 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 different images. In addition to improvement in a learning capacity, GoolgleNet focus was to make CNN parameter efficient. GoogleNet regulates the computation by adding a bottleneck layer using 1x1 convolutional filter, before employing large size kernels. It used sparse connections (not all the output channels are connected to all the input channels), to overcome the problem of redundant information and reduced cost by omitting channels that were not relevant. Furthermore, connections density was reduced by using global average pooling at the last layer, instead of using a fully connected layer. These parameter tunings cause a significant decrease in the number of parameters from 40 million to 5 million parameters. Other regulatory factors applied were batch normalization and use of RmsProp as an optimizer [107]. 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 thus sometimes leads 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 improved

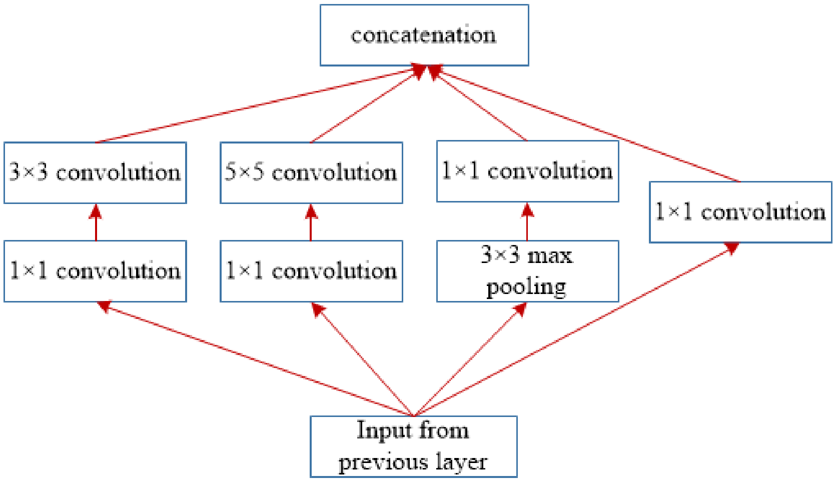

Figure 5: Basic architecture of inception block

feature representations [108]. Network depth has played an important role in the success of supervised training. Theoretical studies have shown that deep networks can represent certain function classes exponentially more efficiently than shallow ones. Csáji [109] represented universal approximation theorem in 2001, which states that a single hidden layer is sufficient to approximate any function but this comes at a cost of exponentially many neurons thus often making it computationally unfeasible. In this regard, Bengio and Delalleau [110] suggested that deeper networks have the potential to maintain the expressive power of the network at a reduced cost [111]. In 2013, Bengio et al. have empirically shown that deep networks are computationally and statistically more efficient for complex tasks [72,112]. Inception and VGG, which showed the best performance in 2014-ILSVR competition, further strengthen the idea that the depth is an essential dimension in regulating learning capacity of networks [24,27,85,86].

4.2.1. Highway Networks

Based on the intuition that the learning capacity can be improved by varying the network depth, Srivastava et al. [87] in 2015, proposed a deep CNN, named as Highway Network. The main problem concerned with deep Nets is slow training and convergence speed. Highway Network exploited depth for learning enriched feature representation by introducing new cross-layer connectivity based training methodology (discussed in Section 4.3.1.). Highway network with

50-layers depth showed better convergence rate than thin and deep architectures on ImageNet dataset [80,81]. Srivastava et al. experimentally showed that performance of plain Net decreases after adding hidden units beyond 10 layers [113]. Highway networks, on the other hand, converge significantly faster than the plain ones even for Nets with 900 layers depth.

4.2.2. ResNet

ResNet was proposed by He et al. [26] which is considered as a continuation of deeper Nets, and it introduced an optimal methodology for the training of deeper Nets (discussed in Section 4.3.2.). ResNet proposed 152-layers deep CNN, which won the 2015-ILSVRC competition. Architecture of residual block is shown in Figure 6. ResNet, which was 20 and 8 times deeper than AlexNet [18] and VGG [24] respectively, showed less computational complexity than previously proposed Nets. He et al. showed ResNet with 50/101/152 layers were more accurate than 34 layers plain Net. Moreover, ResNet gained 28% improvement on famous image recognition benchmark dataset named as COCO [114]. Good performance of ResNet on image recognition and localization tasks depicted that depth is of central importance for many visual recognition tasks.

4.2.3. Inception-V3/4 and Inception-ResNet

Inception-V3/4 and Inception-ResNet, which are an improved version of Inception-V1/2 were proposed by Szegedy et al. [27,85,86]. The idea of Inception-V3 was to reduce the computational cost of deeper Nets without affecting the generalization. For this purpose, Szegedy et al. replaced large size filters (5x5 and 7x7) with small and asymmetric filters (1x7 and 1x5) and used 1x1 convolution as a bottleneck before the large size filters [86]. This makes the traditional convolutional operation more like cross-channel correlation. In one of the previous works, Lin et al. exploited the potential of 1x1 filters in NIN architecture [48]. Szegedy et al. [86] used the same concept in an intelligent way. In Inception-V3, 1x1 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 3x3 or 5x5 convolutions. In Inception-ResNet, Szegedy et al. combined the power of residual learning and inception block [26,27]. In doing so, filter concatenation was replaced by the residual connection. Moreover, the

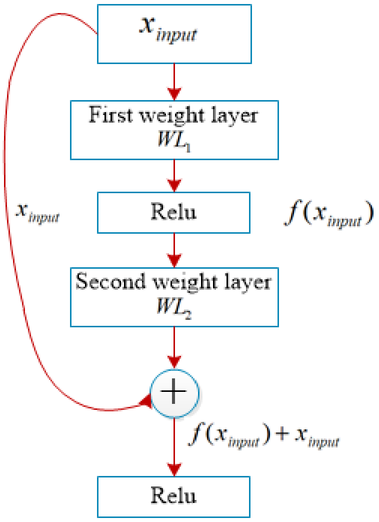

Figure 6: Residual block.

Szegedy et al. experimentally showed that Inception-V4 with residual connections (Inception-ResNet) has same generalization power as Inception-V4 without residual connection with increased depth and width. This clearly depicts that training with residual connections accelerates the training of Inception networks significantly.

4.2.4. ResNext

ResNext, also known as Aggregated Residual Transform Network, is an improvement over the Inception Network [115]. Xie et al. exploited the topology of the split, transform and merge in a powerful but simple way by introducing a new term, cardinality [85]. Cardinality is an additional dimension, which refers to the size of the set of transformations [116,117]. Inception network has not only improved learning capability of conventional CNNs but also makes network resource effective. However, due to the use of diverse spatial embedding's (such as use of 3x3, 5x5 and 1x1 filter) in the transformation branch, each layer needs to be customized separately. ResNext derive characteristic features from Inception, VGG, and ResNet [24,26,85]. ResNext utilized the deep homogenous topology of VGG and simplified GoogleNet architecture by fixing spatial resolution to 3x3 filters within split, transform, and merge block. Building block for ResNext is shown in Figure 7. ResNext used multiple transformations within a split, transform and merge block and defined these transformations in terms of cardinality. Xie et al. showed that increase in cardinality significantly improves the performance. The complexity of ResNext was regulated by applying low embedding's (1x1 filters) before 3x3 convolution. Whereas training was optimized by using skip connections [115].

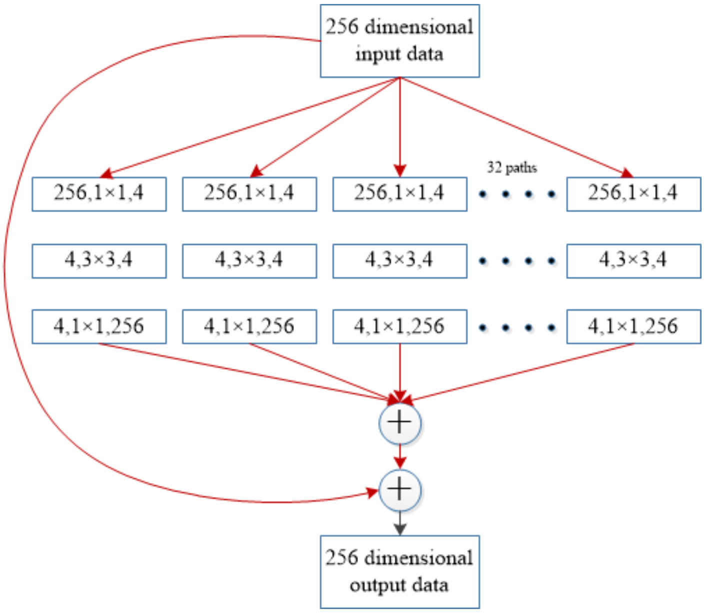

Figure 7: ResNext building block.

4.3. Multi-Path based CNNs

Training of deep networks is challenging and this has been the subject of much recent research on deep Nets. Deep CNN offers both computational and statistical efficiency for complex tasks. However, deeper networks may suffer from performance degradation or gradient vanishing/explosion problems, which are not caused by overfitting but instead by an increase in depth [45,118]. Vanishing gradient problem not only results in higher test error but also in higher training error [118–120]. For the training of deeper Nets, concept of multi-path or cross-layer connectivity was proposed [87,93,94,121]. 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 [122,123]. Cross-layer connectivity partition 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, and 1x1 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 deeper Nets, due to a large number of layers, the backpropagation of error may result in small gradient values at lower layers. To solve this problem, Srivastava et al. [87], in 2015, proposed a new CNN architecture named as Highway Network based on the idea of cross-layer connectivity. In Highway Network, the unimpeded flow of information across layers is enabled by imparting two gating units within a layer (equation (5)). The idea of a gating mechanism was inspired from Long Short Term Memory (LSTM) based Recurrent Neural Networks (RNN) [124,125]. The aggregation of information by combining the l^{th} layer, and previous $l-k$ layers information creates a regularizing effect, making gradient-based training of very deep networks easy. This enables training of a network with more than 100 layers, even as deep as 900 layers with Stochastic Gradient Descent (SGD) algorithm. Cross-layer connectivity for Highway Network is defined in equation (5 and 6).

$$y = H_l(x_i, W_{H_l}).T_g(x_i, W_{T_g}) + x_i \ .C_g(x_i, W_{C_g}) \tag{5}$$

$$C_g(x_i, W_{C_g}) = 1 - T_g(x_i, W_{C_g}) \tag{6}$$

In equation (5), T_g refers to transformation gate, which expresses the amount of the produced output whereas C_g is a carry gate. In a network, $H_l(x_i, W_{H_l})$ shows working of all hidden layers, and shows the residual implementation. Whereas, $1 - T_g(x_i, W_{C_g})$ behaves as a switch in a layer, which decides the path for the flow of information.

4.3.2. ResNet

To address the problem faced during training of deeper Nets, in 2015, He et al. proposed ResNet [26] in which they exploited the idea of bypass pathways used in "Highway Networks". Mathematical formulation of ResNet is expressed in equation (7 & 8).

$$g(x_i) = f(x_i) + x_i \tag{7}$$

$$f(x_i) = g(x_i) - x_i \tag{8}$$

In equation (7), $f(x_i)$ is a transformed signal, whereas x_i is original input. Original input x_i is added to $f(x_i)$ through bypass pathways. In equation (8), $g(x_i) - x_i$, performs residual learning. ResNet introduced shortcut connections within layers to enable cross-layer connectivity but these gates are data independent and parameter free in comparison to 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. ResNet with the depth of 152 layers, (having 20 and 8 times more depth than AlexNet and VGG respectively) won the 2015-ILSVRC championship [18]. Even with increased depth, ResNet has lower computational complexity than VGG [24].

4.3.3. DenseNets

In continuation of Highway Networks and ResNet, DenseNet was proposed to solve the vanishing gradient problem [26,87,93]. The problem with ResNet was that it explicitly preserves information through additive identity transformations due to which many layers contribute very little or no information. Moreover, ResNet has a large number of weights as each layer has a separate set of weights. To address this problem, DenseNet used cross-layer connectivity but, in a modified fashion. DenseNet connected each layer to every other layer in a feed-forward fashion, thus feature maps of all preceding layers were used as inputs into all subsequent layers. This establishes $\frac{l(l+1)}{2}$ direct connections in DenseNet, as compared to l connections between a layer and its preceding layer in traditional CNNs. It imprints the effect of cross-layer depth wise convolutions. As DenseNet concatenates previous layers features instead of adding them, thus, network gain the potential to explicitly differentiate between information that is added to the network and information that is preserved. DenseNet has narrow layer structure; however, it becomes parametrically expensive with an increase in a number of feature maps. The direct admittance of each layer to the gradients through the loss function improves the flow of information throughout the network. This 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 of network architecture was on the power of depth along with the importance of multi-pass regulatory connections in network regularization [26,87]. However, the width of network is as important as depth. Multilayer perceptron gained an advantage of mapping complex functions over perceptron by making parallel use of multiple processing units within a layer. This suggests that width is an important parameter in defining principles of learning along with depth. Lu et al., and Hanin & Sellke have recently shown that NNs with ReLU activation function have to be wide enough in order to hold universal approximation property with the increases in depth [126]. Moreover, 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 [112,127]. Although, stacking of multiple layers (increasing depth) may learn diverse feature representations but not necessarily increase the learning power of the NN. One major problem linked with deep architecture is that some layers or processing units may not learn useful features. To tackle this problem, the focus of research was shifted from deep and narrow architecture towards thin and wide architectures.

4.4.1. WideResNet

The main drawback associated with deep residual networks is feature reuse problem in which some feature transformations or blocks contribute very little to learning. This problem was addressed by WideResNet, proposed by Zagoruyko and Komodakis [28]. Zagoruyko and Komodakis suggested that main learning potential of deep residual networks reside in the residual units, whereas depth has a supplementary effect. WideResNet exploited the power of the residual blocks by making ResNet wide rather than deep [26]. WideResNet increased width by introducing an additional factor *k*, which controls the width of the network. WideResNet showed that the widening of layers provide a much more effective way of performance improvement than making residual networks deep. Although deep residual networks improved representational capacity but they have some demerits such as time intensive training, inactivation of many feature maps (feature reuse problem), and gradient vanishing and exploding problem. He et al. [26], addressed feature reuse problem by incorporating dropout in residual blocks to regularize network in an effective way. Similarly, Huang et al. [91], introduced the concept of stochastic depth by exploiting dropouts to solve vanishing gradient and slow learning problem. The focus of previous research was an increase in depth; therefore, even fraction improvement in performance required the addition of many new layers. An empirical study showed that WideResNet was twice the number of parameters as compared to ResNet but WideResNet can train in a better way than deep networks [28]. Wider residual network was based on the observation that almost all architectures before residual networks, including the most successful Inception and VGG, were wider as compared to ResNet. In WideResNet, learning was made more 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 in AlexNet, VGG, and ResNet, due to deep stacking of multiple convolutional layers, depth of feature maps increase in subsequent layers, whereas spatial dimension decreases, as each convolutional layer is followed by a sub-sampling [18,24,26]. Therefore, in deep CNNs, enriched feature representation is compensated by a decrease in feature map size. The drastic increase in the feature map depth with the loss of spatial information interferes with learning ability. ResNet has shown remarkable results for image classification problem. However, in ResNet the deletion of convolutional block, where dimension of both spatial (spatial dimension decrease) and channel (channel depth increases) changes, usually results in deterioration of the classifier performance. In this regard, stochastic ResNet, improved the performance by reducing information loss associated with the dropping of the residual unit. To address learning interference problem of ResNet, Han et al. proposed, Pyramidal Net [30]. In contrast to the drastic decrease in spatial width with an increase in depth by ResNet, Pyramidal Net increased the width gradually per residual unit in order to cover all possible locations instead of maintaining the same spatial dimension within each residual block until down-sampling appears. Because of a gradual increase in the features map depth, in a top-down fashion, it was named as pyramidal Net. Depth of feature maps is regulated by factor l, which is computed using equation (9).

$$D_l = \begin{cases} 16 & if \;\; l = 1, \\ \left\lfloor D_{l-1} + \dfrac{\lambda}{n} \right\rfloor & if \, 2 \le l \le n+1 \end{cases} \tag{9}$$

D_l denotes the dimension of l^{th} residual unit, n is the total number of the residual units, whereas λ is a step factor and $\frac{\lambda}{n}$ regulates the increase in depth. Depth regulating factor evenly distributes the burden of an increase in feature maps dimension. 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 it results in better generalization [128]. 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 and the other one increases geometrically [43,46]. The major problem is that with the increase in width the quadratic times increase in space and time occurs.

4.4.3. Xception

Xception [129] is an extreme Inception architecture, which exploits the idea of depthwise separable convolution introduced by AlexNet [18]. Xception modified the original inception block by making it wider and replacing different spatial dimensions (1x1, 5x5, 3x3) with a single dimension (1x1 followed by 3x3). Architecture of Xception block is shown in Figure 8. Xception makes the network more computationally efficient by decoupling spatial and channel correlation. It works by first mapping the convolved output to low dimensional embeddings using 1x1 convolution and then spatially transform it k^{th} times. In Xception k is a width defining cardinality, which determines the number of transformations. Xception makes computation easy by separately convolving each channel across spatial axes, which is followed by pointwise convolution (1x1 convolutions) to perform cross-channel correlation. In Xception, 1x1 convolution is used to regulate channel depth. In conventional CNN architectures; conventional convolutional operation uses only one transformation segment, inception block uses three transformation segment whereas in Xception number of transformation segment is equal to a number of channels. Although, the transformation strategy adopted by Xception does not reduce the number of parameters, but it makes learning more efficient and results in improved performance.

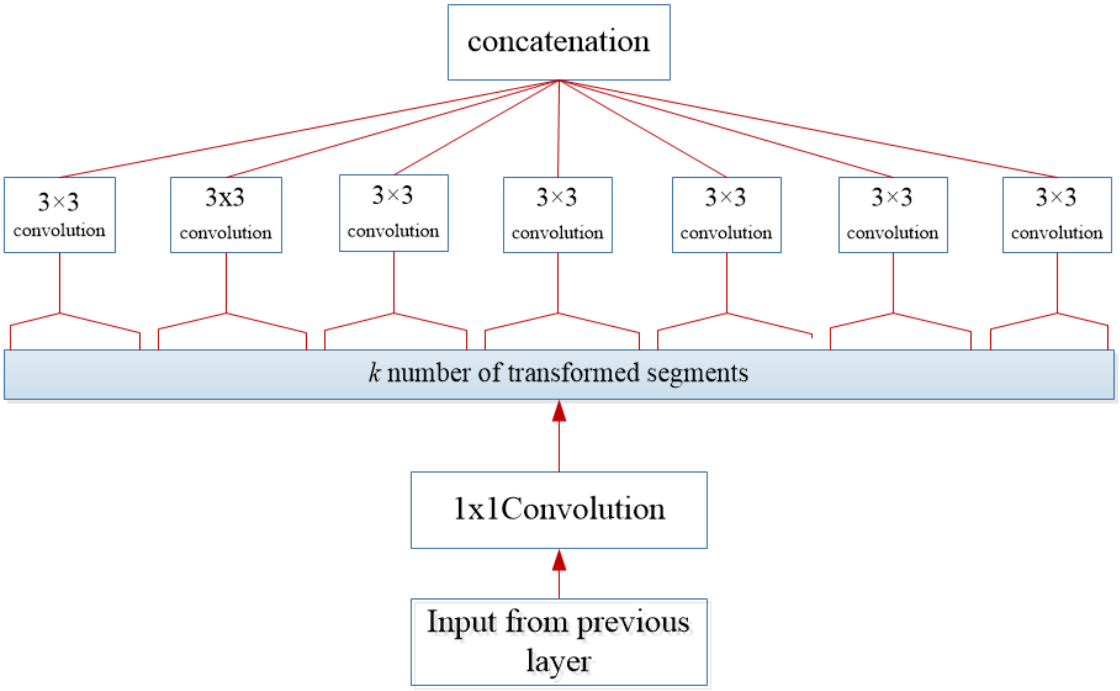

Figure 8: Xception building block.

4.4.4. Inception Family

Inception family of CNNs also comes under the class of width based methods [27,85,86]. In Inception networks, within a layer, varying sizes of filters were used which increased the output of intermediate layers. The use of various sizes of filters are helpful in capturing the diversity in high-level features. Salient characteristics of Inception family are discussed in section 4.1.4 and 4.2.3.

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

CNN got famous for MV tasks because of its hierarchical learning and automatic feature extraction ability [7]. Selection of features play an important role in determining the performance of classification, segmentation, and detection modules. Conventional feature extraction techniques limit the performance of classification module because of a single type of feature [130]. In comparison to conventional techniques, in CNN multiple stages of feature extraction are used, which extract the diverse types of features (known as feature maps in CNN) depending upon the type of input assigned. However, some of the feature maps impart little or no role in object discrimination [131]. Enormous feature sets create an effect of noise and leads towards over-fitting of the network. This suggests that apart from network engineering, selection of class-specific feature maps can play an important role in improving 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., [131] who proposed a new block for the selection of feature maps (commonly known as channels in literature) relevant to object discrimination. This new block was named as SE-block (shown in Figure 9), which suppresses the less important feature maps, but gives high weightage to class specifying feature maps. SE-Network reported record decrease in error on ImageNet dataset. SE-block is a processing unit that was designed in a generic way that can be added in any CNN architecture before the convolution layer. The working of this block is split into two operations; squeeze and excitation. Convolution kernel captures information locally, but it ignores contextual relation of features (correlation) that are outside of this receptive field. To get a global view of feature maps, squeeze block generates channel wise statistics by suppressing spatial information of the convolved input. As global average pooling has the potential to learn the extent of target object effectively, therefore, it was employed by squeeze operation to generate feature map wise statistics using following equation [132].

$$D_M = \frac{1}{m*n}\sum_{i=1}^{m}\sum_{j=1}^{n} x_c(i,j) \qquad (10)$$

D_M is a feature map descriptor, where $m*n$ is a spatial dimension of input. The output of squeeze operation; D_M is assigned to excitation operation, 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 equation (11).

$$V_M = \sigma(w_2\delta(w_1 D_M)) \qquad (11)$$

In equation (11), V_M denotes weightage for each feature map, where δ and σ refers to the ReLU and sigmoid function, respectively. In excitation operation, w_1 and w_2 are used as a regulating factor to limit the model complexity and aid the generalization [43,44]. The output of SE-block was preceded by ReLU activation function, which adds non-linearity in feature maps. Gating mechanism is exploited in SE-block using sigmoid activation function, which models interdependencies among feature map and assigns a weight based on feature map relevance [133]. SE-block is simple in nature and adaptively recalibrates each layer feature maps by multiplying convolved input with the motif responses. These properties give it an edge over other CNN architectures.

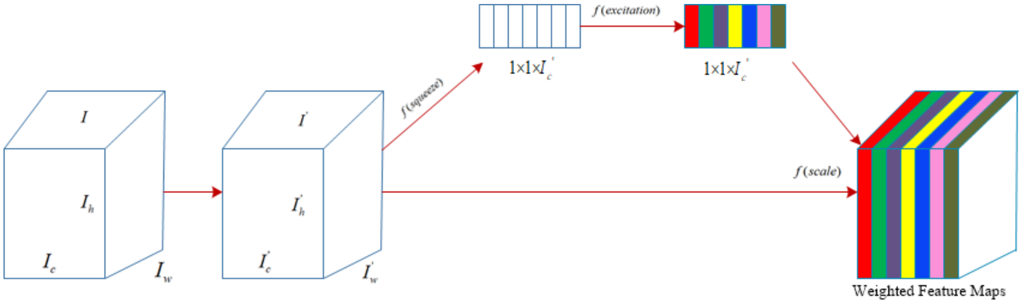

Figure 9: Squeeze and Excitation block.

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. in 2018 [134]. Hu et al. used the idea of SE-block to improve the learning of deep residual networks [26]. 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 channel [131]. This minimizes the impact of SE-block and makes ResNet information redundant. Hu et al. addressed this problem by generating feature map wise statistics from both residual and identity mapping based features. In this regard, global representation of feature maps was generated using global average pooling operation, whereas relevance of feature maps was estimated by making competition between residual and identity mapping based descriptors. This phenomena is termed as inner imaging [134]. CMPE-SE block not only models the relationship between residual feature maps but also maps their relation with identity feature map and makes a competition between residual and identity feature maps. The competition between identity and residual mappings is modeled within block so that the weights of identity channels are larger than the residual channels if identity channels win the competition. The mathematical expression for CMPE-SE block is represented using following equation:

$$y = F_{se}(u_r, x_{id}).F_{res}(x_{id}, w_r) + x_{id} \qquad (12)$$

where x_{id} is the identity mapping of input, F_{se} represents the squeeze operation applied on residual feature map; u_r and identity feature map; x_{id} . F_{res} shows implementation of SE-block on residual feature maps. The output of squeeze operation is multiplied with the SE-block output; F_{res}. The backpropagation algorithm optimizes the competition between identity and residual feature maps and the relationship between all feature maps in the residual block.

4.6. Channel($_{InputChannels}$) Exploitation based CNNs

Image representation plays an important role in determining the performance of the image-processing algorithm. A good representation of the image is one that can define the salient features of an image from a compact code. In different studies, various types of conventional filters are applied to extract different levels of information for a single type of image [135,136]. These diverse representations are used as an input of the model to improve performance [137,138]. CNN is a good feature learner that can automatically extract discriminating features depending upon the problem [139]. However, the learning of CNN relies on input representation. The lack of diversity and absence of class defining information in input affects the CNN performance as a discriminator. For this purpose, the concept of auxiliary learners is introduced in CNN to boost the input representation of the network [31].

4.6.1. Channel Boosted CNN (Channel$_{\text{Input}}$) using TL

In 2018, Khan et al. proposed the 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 so that it can map complex problems [31]. Block diagram of CB-CNN is shown in Figure 10. The channel boosting was performed by artificially creating extra channels (known as auxiliary channels) through deep generative models and then exploiting it through the deep discriminative models. It gives the concept that TL can be used at both generation and discrimination stages. Data representation plays an important role in determining the performance of classifier as different representations explain different levels of information [72]. For improving, the representational potential of the data, Khan et al. exploited the power of TL and deep generative learners [20,140,141]. Generative learners attempt to characterize the data generating distribution during the learning phase, by discovering the set of features or latent variables, which capture variations necessary to generate the structure of the data. In CB-CNN, autoencoders were used as a generative learner 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 input data to original channels space (input channels). CB-CNN encoded channel-boosting phase into a generic block, which was inserted at the start of deep Net. For training, Khan et al. used a pre-trained network, to reduce computational cost. The significance of the study is that two-deep learners are used where generative learning models are used as auxiliary learners, which enhance the representational capacity of deep CNN based discriminator. Although the potential of channel boosting was only evaluated by inserting 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, which may help in improving discrimination power for classes separated by minute differences. CB-CNN is also evaluated on medical image dataset, where it also shows improved results compared to previously proposed approaches. The convergence plot of CB-CNN on mitosis dataset is shown in Figure 11.

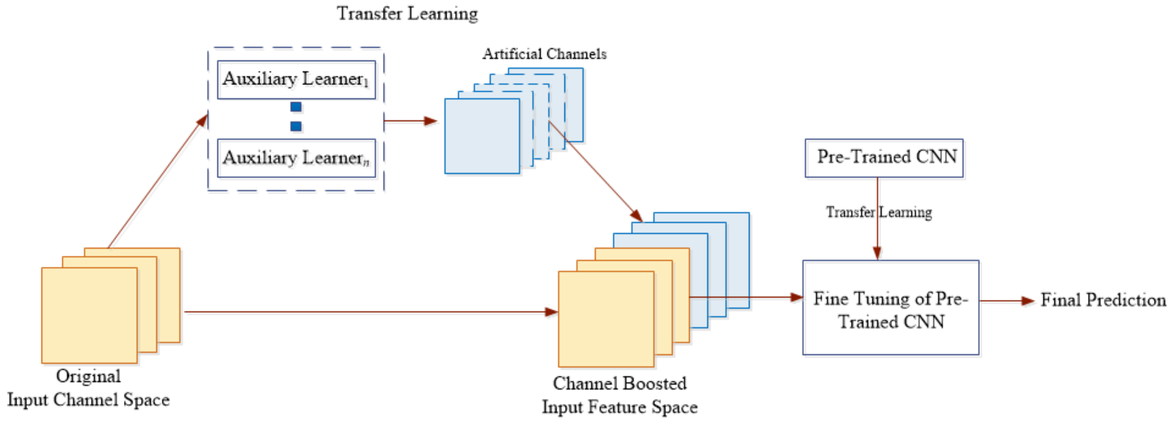

Figure 10: Basic architecture of CB-CNN.

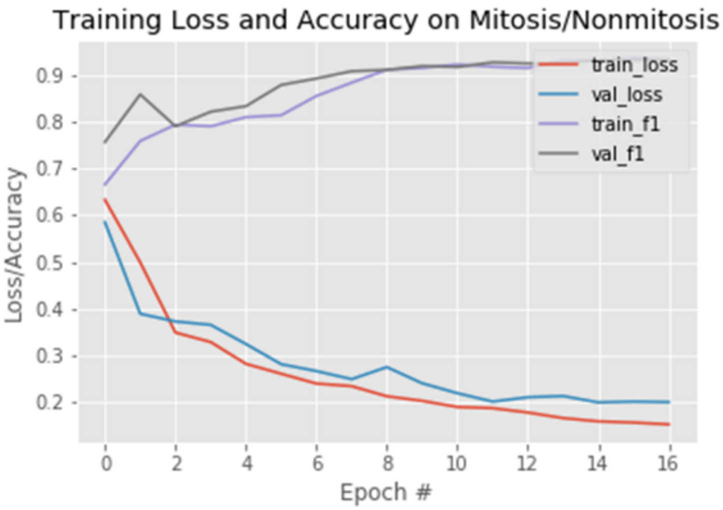

Figure 11: Convergence plot of CB-CNN on mitosis dataset. Loss and accuracy is shown on y-axis, whereas x-axis represents epochs. Training plot of CB-CNN shows that the model converges after about 14 epochs.

4.7. Attention based CNNs

Different levels of abstraction have an important role in defining discrimination power of the NN. In addition to it, selection of features relevant to the context, play a significant role in image localization and recognition. In a human visual system, this phenomenon is referred 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 region but also deduces different interpretations of objects at that location. Thus, it helps to capture visual structure in a better way. Similar kind of interpretability is added into NNs such as RNN and LSTM [124,125]. The aforementioned networks exploited attention module for generation of sequential data and 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 of the data. This idea of attention also helps in making CNN intelligent enough so that it recognizes objects even from cluttered backgrounds and complex scenes.

4.7.1. Residual Attention Neural Network

Wang et al. proposed a Residual Attention Network (RAN) to improve the feature representation of the network [33]. The motivation behind the incorporation of attention in CNN was to make network capable of learning object aware features. RAN is a feed forward CNN, which was built by stacking residual blocks with attention module. Attention module is branched off into trunk and mask branch that adopts 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. Bottom-up feed-forward structure produces low-resolution feature maps with strong semantic information. Whereas, top-down architecture produces dense features in order to make an inference of each pixel. In previously proposed studies, a top-down bottom-up learning strategy was used by Restricted Boltzmann Machines [142]. Goh et al., exploited the top-down attention mechanism as a regularizing factor in Deep Boltzmann Machine (DBM) during the reconstruction phase of the training. Top-down learning strategy optimizes network globally in such a way that output maps to input gradually during the learning process [70,142143]. Attention module in RAN generates object aware soft mask $S_{i,FM}(x_c)$ at each layer [144]. Soft mask, $S_{i,FM}(x_c)$ assigns attention towards object using equation

(13) by recalibrating trunk branch $T_{i,FM}(x_c)$ output thus behaves like a function of a control gate for every neuron output of the trunk branch.

$$A_{i,FM}(x_c) = S_{i,FM}(x_c) * T_{i,FM}(x_c) \tag{13}$$

In one of the previous studies, Transformation network [145,146] also exploited the idea of attention in a simple way by incorporating it with convolution block but the main problem with it was that these 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 layers. 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.

4.7.2. Convolutional Block Attention Module

The significance of attention mechanism and feature map exploitation is validated through RAN and SE-Network [33,96]. In this regard, Woo et al. came up with new attention based CNN named as Convolutional Block Attention Module (CBAM) [32]. CBAM is simple in design similar to SE-Network. SE-Network only considers the contribution of channels in image classification, but it ignores the spatial locality of the object in images. Spatial location of the object has an important role in object detection. CBAM infer attention maps sequentially by applying channel attention followed by spatial attention, in order to find the refined feature maps. In literature, 1x1 convolution and pooling operations are used for spatial attention. Woo et al., showed that pooling of features along spatial axis generates an efficient feature descriptor. CBAM concatenates average pooling operation with max pooling, which generate 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 gives the clue about distinctive object features, whereas use of global average pooling returns suboptimal inference of channel attention. Exploitation of both average pooling and max-pooling improves 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 formulation of 3D attention map via serial learning process helps in reduction of the parameters and computational cost as well. Due to its simplicity, CBAM can be integrated easily with any CNN architecture.

4.7.3. Concurrent Spatial and Channel Excitation Mechanism

Roy et al. extended the work of Hu et al, by incorporating effect of spatial information in combination with channel information to make it applicable to segmentation tasks [96,97]. They introduced three different modules: (i) squeezing spatially and exciting channel-wise (cSE), (ii) squeezing channel-wise and exciting spatially (sSE) and (iii) concurrent spatial and channel squeeze & excitation (scSE). In this work, autoencoder based convolutional NN was used for segmentation, whereas proposed modules were inserted after encoder and decoder layer. In cSE module, the same concept as of SE-block is exploited. In this module, scaling factor is derived based on feature maps contribution in object detection. As spatial information has an important role in segmentation, therefore in sSE module spatial locality has been given more importance than feature map information. For this purpose, different combinations of channels are sliced and exploited spatially to use them for segmentation. In the last module, scSE they assign attention to each channel by deriving scaling factor both from spatial and channel information to selectively highlight the object specific feature maps [97].

5. Applications of CNN

CNN has been successfully applied to the different ML related tasks, namely object detection, recognition, classification, regression, segmentation, etc. However, CNN generally needs a large amount of data for learning. All of the aforementioned areas in which CNN has shown tremendous success have relatively abundant labeled data, such as traffic sign recognition, segmentation of medical images, and the detection of faces, text, pedestrians, and human bodies in natural images. Some applications of CNN are discussed below.

5.1. Natural Language Processing

Natural Language Processing (NLP) converts the language into a presentation that can easily be exploited by any computer. CNN has been applied to NLP based applications such as speech recognition, language modeling, and analysis etc. Especially language modeling or sentence molding has taken twist, after the introduction of CNN as a new representation-learning algorithm. Purpose of sentence modeling is to know the semantics of the sentence for offering new appealing applications according to the customer requirements. Traditional methods of information retrieval analyze the data, based on words or features, but ignore the core of the sentence. In [147], the authors use the dynamic CNN and dynamic *k-max* pooling during training. This approach finds the word relations without taking into account any external source like parser or vocabulary. In the similar way, collobert et al. [148] proposed CNN based architecture that can perform various MLP related tasks at the same time like chunking, language modeling, recognizing name-entity and role modeling related to semantics. In another work, Hu et al. [149] proposed a generic CNN based architecture that performs matching between two sentences and thus can be applied to different languages.

5.2. Activity Recognition

Activity recognition is associated with recognition of different actions and activities. Face detection, pose estimation, action recognition, etc. are the different areas that come under activity recognition. Face detection is one of the challenging tasks in human activity recognition. The recent research on face recognition is working to cope with the challenges that put the original image into big variations even when they do not exist in reality. This variation is caused by illumination, change in pose, and different facial expressions. Farfade et al. [150] proposed deep CNN for detecting face from different pose and also able to recognize occluded faces. In another work, Zhang et al. [151] performed face detection using a new type of multitask cascaded CNN. Zhang's technique showed good results when comparison is shown against latest state-of-the-art techniques [152–154]. Human pose estimation is another tedious task related to MV because of the high variability of body pose. Li et al. [155] proposed a heterogeneous deep CNN based pose estimation related technique. In Li's technique, empirical results have shown that the hidden neurons are able to learn the localized part of the body after performing parameter tuning. Similarly, another cascade based CNN technique is proposed by Bulat et al. [156]. In their cascaded architecture, first heat maps were detected, whereas, in the second phase, regression was performed on the detected heat maps.

Action recognition also comes under 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, use of Hidden Markov Models or more recently action sketch generation. Recently, Wang et al. [157] proposed a three dimensional CNN architecture along with LSTM for recognizing different actions from video frames. Experimental results have shown that Wang's technique outperforms the latest activity recognition based techniques [158–162]. Similarly, another three dimensional CNN based action recognition system is proposed by Ji et al. [163]. 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 supervised way and able to perform activity recognition in real world applications.

5.3. Object Detection

Object detection is considered as a task of identifying different objects from the images. Recently, region-based CNN (R-CNN) has been widely used for object detection. Ren et al. [164] proposed an improvement over R-CNN named as fast R-CNN for object detection. In the proposed work full convolution is used to extract feature space that can simultaneously detect boundary and score of object located at different positions. Similarly, Dai et al. [165] 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 proposed by Gidaris et al. [166], 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 the 2012 challenge dataset.

5.4. Image Classification

CNN has been widely used for images classification [167–169]. One of the major applications of CNN is in medical images especially, for diagnoses of cancer using histopathological images [170]. Recently, Spanhol et al. [171] used CNN for the diagnosis of breast cancer images and results are compared against the network that is trained on dataset that contain handcrafted feature descriptors [172]. Another recently proposed CNN based technique for breast cancer diagnosis is developed by Wahab et al. [173]. In Wahab's work, two phases are involved. In the first phase, hard non-mitosis examples are identified. Whereas, in second phase data augmentation is performed to cope with the class skewness problem in breast cancer based histopathological images. Similarly, Ciresan et al. [82] used German benchmark dataset related to traffic sign signal for designing CNN based architecture that performed traffic sign classification related task with good recognition rate.

5.5. Speech Recognition

Speech is considered as a communication link between human beings. In the ML field, before the availability of hardware resources, speech recognition models do not show good results. With the advancement in hardware resources, training of DNN with a lot of training data becomes possible. Deep CNN is always considered best for image classification, however, recent studies have shown that it also performs better on speech recognition related tasks. Hamid et al. [174] proposed CNN based speaker independent speech recognition system. Experimental results showed ten percent reduction in error rate in comparison to error reported by the previously proposed methods [175,176]. In another work [177], various CNN architectures, which are either based on the full or limited number of weight sharing within the convolution layer, are explored. Furthermore, the performance of the CNN is also checked after the initialization of whole network using pre-training phase [175]. Experimental results showed that all of the explored architectures show good performance on phone and vocabulary recognition related tasks.

6. CNN Challenges

Deep CNN has achieved good performance on data that either is of the time series nature or follows a grid like topology. However, there are also some challenges that are associated with the use of deep CNN architecture for ML related tasks. 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 leads to improved performance. Similarly, Hinton mentioned that lower layers should handover its knowledge only to the relevant neurons of the next layer. In this regard, Hinton proposed the Capsule Network approach [178,179].

In another work, Szegedy et al. showed that training of CNN architecture on noisy image data can cause an increase of misclassification error [180]. The addition of the small quantity of random noise in the input image is capable to fool the network in such a way that, the model will classify the original and its slightly perturbed version differently.

Despite of all the challenges reported by the different researchers related to performance of CNN on different ML tasks. There are some challenges faced during the training of deep CNN model; some of which are given below:

- Deep NN are generally like a black box and thus lack interpretability and explainability. Therefore, sometimes it is difficult to verify them, and in case of vision related tasks, CNN may offer little robustness against noise and other alterations to images.
- Each layer of CNN automatically extracts better and problem specific features related to task. However, for some tasks, it is important to know the nature of features extracted by deep CNN before the classification. The idea of feature visualization in CNNs can help in this direction.
- Deep CNNs are based on supervised learning mechanism, and therefore, availability of a large and annotated data is required for its proper learning. In contrast, humans have the ability to learn and generalize from a few examples.

- Hyperparameter selection highly influences the performance of CNN. A little change in the hyperparameter values can affect the overall performance of a CNN. That is why careful selection of parameters is a major design issue that needs to be addressed through some optimization strategy.
- Efficient training of CNN demands powerful hardware resources such as GPUs. However, it is still needed to explore that how to efficiently employ CNN in embedded and smart devices. Few applications of deep learning in embedded systems are wound intensity correction [181], law enforcement in smart cities [182], and many other [183].

7. Future Directions

The exploitation of different innovative ideas in CNN architectural design has changed the direction of research in computer vision. The magnificent performance of CNN on grid like topological data present it as a powerful representation model for image data. CNN is an area that is developing with ongoing progress and it is likely to continue and sustain its growth in future as well.

- Ensemble learning [184] is one of the prospective areas of research in CNNs. The combination of multiple and diverse architectures can aid model in improving generalization on diverse categories of images by extracting different levels of semantic image representation. Similarly, concepts such as batch normalization, dropout, and new activation functions are also worth mentioning.
- The potential of CNN as a generative learner is exploited in image segmentation tasks where it has shown good results [185]. The exploitation of generative learning capabilities of CNN at supervised feature extraction stages (learning of filter using backpropagation) can boost the representation power of the model. Now new paradigms are needed that can enhance the learning capacity of CNN by incorporating the informative feature maps that are learnt using the auxiliary at the intermediate feature learning stages of CNN [31].
- In human visual system, attention is one of the important mechanisms in capturing information from images. Attention mechanism operates in such a way that it not only extract the essential information from image but also store its contextual relation with

other components of images and its spatial relevance [186,187]. In future, research will be carried out in the direction that preserves the spatial relevance of object along with discriminating features of object at later stages of learning.

- The learning capacity of CNN is enhanced by exploiting the size of the network and it was made possible with the advancement in hardware processing units and computational resources. However, the training of deep and high capacity architectures is a significant overhead on memory usage and computational resources. This requires a lot of improvements in hardware that can accelerate research in CNNs. The main concern with CNNs is the run-time applicability. Moreover, 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 [188]. Some of the accelerators are already proposed, such as Application Specific Integrated Circuits, Eyeriss and Google Tensor Processing Unit [189]. Moreover, different operations have been performed to save hardware resources in terms of chip area and power, by reducing precision of operands and ternary quantization, or reducing the number of matrix multiplication operations. Now it is also time to redirect research towards hardware-oriented approximation models [190].
- Deep CNN has large number of parameters such as activation function, kernel size, number of neurons per layers, and arrangement of layers etc. The selection of hyperparameters depending upon the problem at hand and its evaluation time makes parameter tuning quite difficult in the context of deep learning. Hyper-parameter tuning is a tedious and intuition driven task, which cannot be defined via explicit formulation. Genetic algorithms can also be used to automatically optimize the hyper-parameter by performing search both in a random fashion as well as by directing search by utilizing previous results [191–193].
- The learning capacity of deep CNN model has a strong correlation with the size of the model. However, capacity of deep CNN model is restricted due to hardware resources. 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 machine learning library; GPipe[194] that uses synchronous stochastic gradient descent and pipeline parallelism for training. In future, the concept of pipelining can be used to

accelerate the training of large models and to scale the performance without tuning hyperparameters.

8. Conclusion

CNN has made remarkable progress in image processing and has revived interest in ANNs. Up till now, a lot of research has been carried out to improve the CNN's performance on vision related tasks. The advancements in CNNs can be categorized in different ways including activation, and loss function, optimization, regularization, learning algorithms, and restructuring of processing units. This paper reviews advancements in the CNN architectures, and based on the design patterns of processing units, it has proposed the taxonomy for CNN architectures. In addition to categorization of CNNs into different classes, this paper also covers evolutionary history of CNNs, its applications, challenges, and future directions.

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. Now the new paradigm of research in CNN architectures is mainly the development of new and effective block architectures. The role of these blocks in a network is that of an auxiliary learner, which by either exploiting spatial and feature map information or boosting of input channels improves the overall performance. These blocks play a significant role in boosting of CNN performance by making problem aware learning. Moreover, block based architecture of CNN encourages learning in a modular fashion thereby making architecture more simple and understandable. The concept of block being a structural unit is going to persist and further enhance CNN performance. Additionally, the idea of exploiting channel information in addition to spatial information within a block will gain more importance.

Table 1: Performance comparison of recent architectures of different categories. Top 5 error rate is reported for all architectures.

Architecture Name	Year	Main contribution	Parameters	Error Rate	Depth	Category	Researcher
LeNet [56]	1998	- First Popular CNN architecture	0.060 M	[dist]MNIST: 0.8 MNIST: 0.95	7	Spatial Exploitation	LeCun, Y. *et al.*
AlexNet [18]	2012	- Deeper and wider than the LeNet - Uses Relu ,Dropout and overlap Pooling - GPUs NVIDIA GTX 580	60 M	ImageNet: 15.3	8	Spatial Exploitation	Krizhevsky, A. *et al.*
ZefNet [23]	2014	- Intermediate layers feature visualization	60 M	ImageNet: 14.8	8	Spatial Exploitation	Zeiler & Fergus
VGG [24]	2014	- Homogenous topology - Small kernel size	138 M	ImageNet: 7.3	16	Spatial Exploitation	Simonyan & Zisserman
GoogLeNet [85]	2015	- Split Transform Merge - Introduces block concept	4 M	ImageNet: 6.67	22	Spatial Exploitation	Szegedy, C. *et al.*
Inception-V3 [86]	2015	- Handles the problem of a representational bottleneck - Replace large size filters with small filters - Replaces the bigger filter with smaller filters	23.6 M	ImageNet:- Multi-Crop:3.58 Single-Crop:5.6	-	Depth	Szegedy, C. *et al.*
Highway Networks [87]	2015	- Multi-Path Idea	2.3 M	CIFAR-10: 7.76	19	Depth + Multi-Path	Srivastava, R. K. *et al.*
Inception-V4 [86]	2016	- Split, Transform, Merge Uses asymmetric filter	-	ImageNet: 3.8	-	Depth	Szegedy.C. *et al.*
Inception-ResNet [86]	2016	- Split, Transform, Merge and Residual Links	-	ImageNet: 3.1		Depth + Multi-Path	Szegedy.C. *et al.*
ResNet [26]	2016	- Residual Learning and Identity mapping based skip connection	6.8 M 1.7 M	ImageNet: 3.57 CIFAR-10: 6.43	152 110	Spatial Exploitation + Depth + Multi-Path	He, K. *et al.*
DelugeNet [94]	2016	- Allow cross layer information inflow in Deep Networks	20.2 M	CIFAR-10: 3.76 CIFAR-100: 19.02	146	Multi-path	Kuen, J. *et al.*
FractalNet [121]	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, G. *et al.*
WideResNet [28]	2016	- Width is increased and depth is decreased	36.5 M	CIFAR-10: 3.89 CIFAR-100: 18.85	28 -	Width	Zagoruyko & Komodakis
Xception [129]	2017	- Depth wise Convolution followed by point wise convolution	22.8 M	ImageNet: 0.055	36	Width	Chollet, F.
Residual Attention Neural Network [33]	2017	- Introduces Attention Mechanism	8.6 M	CIFAR-10: 3.90 CIFAR-100: 20.4 ImageNet: 4.8	452	Attention	Wang, F. *et al.*
ResNexT [115]	2017	- Cardinality - Homogeneous topology - Grouped convolution	68.1 M	CIFAR-10: 3.58 CIFAR-100: 17.31 ImageNet: 4.4	29 101	Spatial Exploitation	Xie, S. *et al*.
Squeeze & Excitation Networks [131]	2017	- Models Interdependencies between feature maps	-	ImageNet: 3.58	154	Feature Map Exploitation	Hu, J. *et al.*
DenseNet [93]	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, G. *et al.*
PolyNet [195]	2017	- Experimented structural diversity - Introduces Poly Inception Module - Generalizes residual unit using Polynomial compositions	-	ImageNet: Single:4.25 Multi:3.45	- -	Width	Zhang, X. *et al.*
PyramidalNet [30]	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. D. *et al.*
Convolutional Block Attention Module (ResNeXt101 (32x4d) + CBAM) [32]	2018	- Exploit both spatial and feature map information	48.96 M	ImageNet: 5.59	101	Attention	Woo. S. *et al.*
Concurrent Squeeze & Channel Excitation Mechanism [97]	2018	- Squeezing spatially followed by exciting channel-wise - Squeezing channel-wise followed by exciting spatially - Performing spatial and channel squeeze & excitation in parallel	-	MALC: 0.12 Visceral: 0.09	-	Attention	Guha. A. *et al.*
Channel Boosted CNN [31]	2018	- Boost the original channels with extra generated information rich channels	-	-	-	Channel Boosted	Khan. A. *et al.*
Competitive Squeeze & Excitation Network CMPE-SE-WRN-28 [134]	2018	- Residual and identity mappings both are responsible for rescaling the channel	36.92 M 36.90 M	CIFAR-10: 3.58 CIFAR-100: 18.47	28 28	Feature Map Exploitation	Hu. Y. *et al.*

Acknowledgments

We thank Pattern Recognition lab at DCIS, and PIEAS for providing us computational facilities.

References

1. Chapelle, O. Support vector machines for image classification. *Stage deuxième année magistère d'informatique l'École Norm. Supérieur Lyon* **10**, 1055–1064 (1998).

2. Lowe, D. G. Object recognition from local scale-invariant features. *Proc. Seventh IEEE Int. Conf. Comput. Vis.* 1150–1157 vol.2 (1999). doi:10.1109/ICCV.1999.790410

3. Bay, H., Ess, A., Tuytelaars, T. & Van Gool, L. Speeded-Up Robust Features (SURF). *Comput. Vis. Image Underst.* **110**, 346–359 (2008).

4. Dalal, N. & Triggs, W. Histograms of Oriented Gradients for Human Detection. *2005 IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. CVPR05* **1**, 886–893 (2004).

5. Ojala, T., Pietikäinen, M. & Harwood, D. A comparative study of texture measures with classification based on feature distributions. *Pattern Recognit.* **29**, 51–59 (1996).

6. Heikkilä, M., Pietikäinen, M. & Schmid, C. Description of interest regions with local binary patterns. *Pattern Recognit.* **42**, 425–436 (2009).

7. Lecun, Y. & Kavukcuoglu, K. IEEE Xplore - Convolutional networks and applications in vision. doi:10.1109/ISCAS.2010.5537907

8. Wiesel, T. N. <Hubel_et_al-1968-The_Journal_of_Physiology.pdf>. 215–243 (1968).

9. Ian Goodfellow, Bengio, Y. & Courville, A. Deep learning. *Nat. Methods* **13**, 35 (2017).

10. LeCun, Y. *et al.* Backpropagation applied to handwritten zip code recognition. *Neural Comput.* **1**, 541–551 (1989).

11. Ciresan, D., Giusti, A., Gambardella, L. M. & Schmidhuber, J. Deep neural networks segment neuronal membranes in electron microscopy images. in *Advances in neural information processing systems* 2843–2851 (2012).

12. Deng, L., Yu, D. & Delft, B. —. Deep Learning: Methods and Applications Foundations and Trends R in Signal Processing. *Signal Processing* **7**, 3–4 (2013).

13. Jarrett, K., Kavukcuoglu, K., Ranzato, M. & LeCun, Y. What is the best multi-stage

architecture for object recognition? BT - Computer Vision, 2009 IEEE 12th International Conference on. *Comput. Vision, 2009 …* 2146–2153 (2009).

14. Scherer, D., Müller, A. & Behnke, S. Evaluation of pooling operations in convolutional architectures for object recognition. in *Artificial Neural Networks--ICANN 2010* 92–101 (Springer, 2010).

15. Bengio, Y. Learning Deep Architectures for AI. *Found. Trends® Mach. Learn.* **2**, 1–127 (2009).

16. Laskar, M. N. U., Giraldo, L. G. S. & Schwartz, O. Correspondence of Deep Neural Networks and the Brain for Visual Textures. 1–17 (2018).

17. Grill-Spector, K., Weiner, K. S., Gomez, J., Stigliani, A. & Natu, V. S. The functional neuroanatomy of face perception: From brain measurements to deep neural networks. *Interface Focus* **8**, (2018).

18. Krizhevsky, A., Sutskever, I. & Hinton, G. E. ImageNet Classification with Deep Convolutional Neural Networks. *Adv. Neural Inf. Process. Syst.* 1–9 (2012). doi:http://dx.doi.org/10.1016/j.protcy.2014.09.007

19. Qureshi, A. S., Khan, A., Zameer, A. & Usman, A. Wind power prediction using deep neural network based meta regression and transfer learning. *Appl. Soft Comput. J.* **58**, 742–755 (2017).

20. Qiang Yang, Pan, S. J. & Yang, Q. A Survey on Transfer Learning. **1**, 1–15 (2008).

21. Guo, Y. *et al.* Deep learning for visual understanding: A review. *Neurocomputing* **187**, 27–48 (2016).

22. Liu, W. *et al.* A survey of deep neural network architectures and their applications. *Neurocomputing* **234**, 11–26 (2017).

23. Zeiler, M. D. & Fergus, R. Visualizing and Understanding Convolutional Networks. *arXiv Prepr. arXiv1311.2901v3* **30**, 225–231 (2013).

24. Simonyan, K. & Zisserman, A. VERY DEEP CONVOLUTIONAL NETWORKS FOR LARGE-SCALE IMAGE RECOGNITION. *ICLR* **75**, 398–406 (2015).

25. Szegedy, C., Liu, W. & Jia, Y. Going deeper with convolutions. *arXiv Prepr.*

arXiv1409.4842v1 **57**, 62–63 (2014).

26. He, K., Zhang, X., Ren, S. & Sun, J. Deep Residual Learning for Image Recognition. *Multimed. Tools Appl.* **77**, 10437–10453 (2015).

27. Szegedy, C., Ioffe, S. & Vanhoucke, V. Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning. *arXiv Prepr. arXiv1602.07261v2* **131**, 262–263 (2016).

28. Zagoruyko, S. & Komodakis, N. Wide Residual Networks. (2016).

29. Xie, S., Girshick, R., Dollár, P., Tu, Z. & He, K. Aggregated Residual Transformations for Deep Neural Networks. (2016). doi:10.1109/CVPR.2017.634

30. Han, D., Kim, J. & Kim, J. Deep pyramidal residual networks. *Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017* **2017–Janua**, 6307–6315 (2017).

31. Khan, A., Sohail, A. & Ali, A. A New Channel Boosted Convolutional Neural Network using Transfer Learning. (2018).

32. Woo, S., Park, J., Lee, J.-Y. & Kweon, I. S. CBAM: Convolutional Block Attention Module. (2018).

33. Wang, F. *et al.* Residual attention network for image classification. *Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017* **2017–Janua**, 6450–6458 (2017).

34. Gu, J. *et al.* Recent advances in convolutional neural networks. *Pattern Recognit.* **77**, 354–377 (2018).

35. Zhang, Q. *et al.* Recent advances in convolutional neural network acceleration. *Neurocomputing* **323**, 37–51 (2019).

36. Najafabadi, M. M. *et al.* Deep learning applications and challenges in big data analytics. *J. Big Data* **2**, 1–21 (2015).

37. Bouvrie, J. 1 Introduction Notes on Convolutional Neural Networks. (2006). doi:http://dx.doi.org/10.1016/j.protcy.2014.09.007

38. Lecun, Y., Bengio, Y. & Hinton, G. Deep learning. *Nature* **521**, 436–444 (2015).

39. Lee, C.-Y., Gallagher, P. W. & Tu, Z. Generalizing pooling functions in convolutional neural networks: Mixed, gated, and tree. in *Artificial Intelligence and Statistics* 464–472

(2016).

40. Huang, F. J., Boureau, Y.-L., LeCun, Y. & others. Unsupervised learning of invariant feature hierarchies with applications to object recognition. in *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on* 1–8 (2007).

41. Wang, T., Wu, D. J. D. J., Coates, A. & Ng, A. Y. End-to-end text recognition with convolutional neural networks. *ICPR, Int. Conf. Pattern Recognit.* 3304–3308 (2012).

42. Boureau, Y. Icml2010B.Pdf. (2009). doi:citeulike-article-id:8496352

43. Xu, B., Wang, N., Chen, T. & Li, M. Empirical Evaluation of Rectified Activations in Convolutional Network. (2015). doi:10.1186/1757-1146-1-S1-O22

44. Dalgleish, T. *et al.* [No Title]. *J. Exp. Psychol. Gen.* **136**, 23–42 (2007).

45. Hochreiter, S. The vanishing gradient problem during learning recurrent neural nets and problem solutions. *Int. J. Uncertainty, Fuzziness Knowledge-Based Syst.* **6**, 107–116 (1998).

46. Ioffe, S. & Szegedy, C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. (2015). doi:10.1016/j.molstruc.2016.12.061

47. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. & Salakhutdinov, R. Dropout: A Simple Way to Prevent Neural Networks from Overfittin. *J. Mach. Learn. Res.* **1**, 11 (2014).

48. Lin, M., Chen, Q. & Yan, S. Network In Network. 1–10 (2013). doi:10.1109/ASRU.2015.7404828

49. Rawat, W. & Wang, Z. Deep Convolutional Neural Networks for Image Classification: A Comprehensive Review. **61**, 1120–1132 (2016).

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

51. Fukushima, K. Neocognitron: A hierarchical neural network capable of visual pattern recognition. *Neural networks* **1**, 119–130 (1988).

52. Linnainmaa, S. 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 (1970).

53. Zhang, X. & LeCun, Y. Text understanding from scratch. *arXiv Prepr. arXiv1502.01710* (2015).

54. LeCun, Y., Bottou, L., Bengio, Y. & Haffner, P. Gradient-based learning applied to document recognition. *Proc. IEEE* **86**, 2278–2324 (1998).

55. LeCun, Y. & others. LeNet-5, convolutional neural networks. *URL http//yann. lecun. com/exdb/lenet* 20 (2015).

56. LeCun, Y. *et al.* Learning algorithms for classification: A comparison on handwritten digit recognition. *Neural networks Stat. Mech. Perspect.* **261**, 276 (1995).

57. Joachims, T. Text categorization with support vector machines: Learning with many relevant features. in *European conference on machine learning* 137–142 (1998).

58. Decoste, D. & Schölkopf, B. Training invariant support vector machines. *Mach. Learn.* **46**, 161–190 (2002).

59. Liu, C.-L., Nakashima, K., Sako, H. & Fujisawa, H. Handwritten digit recognition: benchmarking of state-of-the-art techniques. *Pattern Recognit.* **36**, 2271–2285 (2003).

60. Schmidhuber, J. New millennium AI and the convergence of history. in *Challenges for computational intelligence* 15–35 (Springer, 2007).

61. Simard, P. Y., Steinkraus, D. & Platt, J. C. Best practices for convolutional neural networks applied to visual document analysis. in *null* 958 (2003).

62. Deng, L. The MNIST database of handwritten digit images for machine learning research [best of the web]. *IEEE Signal Process. Mag.* **29**, 141–142 (2012).

63. Chellapilla, K., Puri, S. & Simard, P. High performance convolutional neural networks for document processing. in *Tenth International Workshop on Frontiers in Handwriting Recognition* (2006).

64. Abdulkader, A. Two-tier approach for Arabic offline handwriting recognition. in *Tenth International Workshop on Frontiers in Handwriting Recognition* (2006).

65. Cire\csan, D. C., Meier, U., Gambardella, L. M. & Schmidhuber, J. Deep, big, simple

neural nets for handwritten digit recognition. *Neural Comput.* **22**, 3207–3220 (2010).

66. LeCun, Y., Kavukcuoglu, K. & Farabet, C. Convolutional networks and applications in vision. *ISCAS 2010 - 2010 IEEE Int. Symp. Circuits Syst. Nano-Bio Circuit Fabr. Syst.* 253–256 (2010). doi:10.1109/ISCAS.2010.5537907

67. Matsugu, M., Mori, K., Ishii, M. & Mitarai, Y. Convolutional spiking neural network model for robust face detection. in *Neural Information Processing, 2002. ICONIP'02. Proceedings of the 9th International Conference on* **2**, 660–664 (2002).

68. Chen, Y.-N., Han, C.-C., Wang, C.-T., Jeng, B.-S. & Fan, K.-C. The application of a convolution neural network on face and license plate detection. in *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on* **3**, 552–555 (2006).

69. Fasel, B. Facial expression analysis using shape and motion information extracted by convolutional neural networks. in *Neural Networks for Signal Processing, 2002. Proceedings of the 2002 12th IEEE Workshop on* 607–616 (2002).

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

71. Bengio, Y., Lamblin, P., Popovici, D. & Larochelle, H. Greedy Layer-Wise Training of Deep Networks.

72. Bengio, Y., Courville, A. & Vincent, P. Representation learning: A review and new perspectives. *IEEE Trans. Pattern Anal. Mach. Intell.* **35**, 1798–1828 (2013).

73. Giusti, A., Cire\csan, D. C., Masci, J., Gambardella, L. M. & Schmidhuber, J. Fast image scanning with deep max-pooling convolutional neural networks. in *2013 IEEE International Conference on Image Processing* 4034–4038 (2013).

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

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

76. Cire\csan, D. C., Meier, U., Masci, J., Gambardella, L. M. & Schmidhuber, J. High-

performance neural networks for visual object classification. *arXiv Prepr. arXiv1102.0183* (2011).

77. Nickolls, J., Buck, I., Garland, M. & Skadron, K. Scalable parallel programming with CUDA. in *ACM SIGGRAPH 2008 classes* 16 (2008).

78. Lindholm.Erik, Oberman.Stuart, Montrym.John, N. J. NVIDIA TESLA : A Unified Graphics And Computing Architecture Computing Architecture. Its Scalable Parallel Aarray Of Processor Is. *Ieee Micro* 39–55 (2008). doi:10.1109/MM.2008.31

79. Cire\csan, D., Meier, U. & Schmidhuber, J. Multi-column deep neural networks for image classification. *arXiv Prepr. arXiv1202.2745* (2012).

80. Morar, A., Moldoveanu, F. & Gröller, E. Image segmentation based on active contours without edges. *Proc. - 2012 IEEE 8th Int. Conf. Intell. Comput. Commun. Process. ICCP 2012* 213–220 (2012). doi:10.1109/ICCP.2012.6356188

81. Berg, A., Deng, J. & Fei-Fei, L. Large scale visual recognition challenge 2010. (2010).

82. Cireşan, D., Meier, U., Masci, J. & Schmidhuber, J. Multi-column deep neural network for traffic sign classification. *Neural Networks* **32**, 333–338 (2012).

83. Sinha, T., Verma, B. & Haidar, A. Optimization of convolutional neural network parameters for image classification. *2017 IEEE Symp. Ser. Comput. Intell. SSCI 2017 - Proc.* **2018**–**Janua**, 1–7 (2018).

84. Alom, M. Z. *et al.* The History Began from AlexNet: A Comprehensive Survey on Deep Learning Approaches. *Desalination* **5**, 293–329 (2018).

85. Szegedy, C. *et al.* Going Deeper with Convolutions. *arXiv:1409.4842* (2014). doi:10.1109/CVPR.2015.7298594

86. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J. & Wojna, Z. Rethinking the Inception Architecture for Computer Vision. in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* 2818–2826 (IEEE, 2016). doi:10.1109/CVPR.2016.308

87. Srivastava, R. K., Greff, K. & Schmidhuber, J. Highway Networks. (2015). doi:10.1002/esp.3417

88. Chung, J., Gulcehre, C., Cho, K. & Bengio, Y. Empirical Evaluation of Gated Recurrent

Neural Networks on Sequence Modeling. *2015 IEEE Int. Conf. Rehabil. Robot.* **2015–Septe**, 119–124 (2014).

89. Karpathy, A., Johnson, J. & Fei-Fei, L. Visualizing and Understanding Recurrent Networks. 1–12 (2015). doi:10.1007/978-3-319-10590-1_53

90. Yamada, Y., Iwamura, M. & Kise, K. Deep pyramidal residual networks with separated stochastic depth. *arXiv Prepr. arXiv1612.01230* (2016).

91. Huang, G., Sun, Y., Liu, Z., Sedra, D. & Weinberger, K. Q. Deep networks with stochastic depth. in *European Conference on Computer Vision* 646–661 (Springer, 2016).

92. Targ, S., Almeida, D. & Lyman, K. Resnet in Resnet: generalizing residual architectures. *arXiv Prepr. arXiv1603.08029* (2016).

93. Huang, G., Liu, Z., Van Der Maaten, L. & Weinberger, K. Q. Densely connected convolutional networks. *Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017* **2017–Janua**, 2261–2269 (2017).

94. Kuen, J., Kong, X., Wang, G. & Tan, Y. P. DelugeNets: Deep Networks with Efficient and Flexible Cross-Layer Information Inflows. *Proc. - 2017 IEEE Int. Conf. Comput. Vis. Work. ICCVW 2017* **2018–Janua**, 958–966 (2018).

95. Kuen, J., Kong, X., Wang, G., Tan, Y.-P. & Group, A. DelugeNets: Deep Networks with Efficient and Flexible Cross-layer Information Inflows. in *Computer Vision Workshop (ICCVW), 2017 IEEE International Conference on* 958–966 (2017).

96. Hu, J., Shen, L., Albanie, S., Sun, G. & Wu, E. Squeeze-and-Excitation Networks. 1–14 (2017). doi:10.1108/15736101311329151

97. Roy, A. G., Navab, N. & Wachinger, C. Concurrent spatial and channel 'squeeze & excitation' in fully convolutional networks. *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)* **11070 LNCS**, 421–429 (2018).

98. Shin, H.-C. *et al.* Deep Convolutional Neural Networks for Computer-Aided Detection: CNN Architectures, Dataset Characteristics and Transfer Learning. *IEEE Trans. Med. Imaging* **35**, 1285–1298 (2016).

99. Kafi, M., Maleki, M. & Davoodian, N. 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 (2015).

100. Potluri, S., Fasih, A., Vutukuru, L. K., Al Machot, F. & Kyamakya, K. CNN based high performance computing for real time image processing on GPU. in *Proceedings of the Joint INDS'11 & ISTET'11* 1–7 (2011).

101. Gardner, M. W. & Dorling, S. R. Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences. *Atmos. Environ.* **32**, 2627–2636 (1998).

102. Dahl, G. E., Sainath, T. N. & Hinton, G. E. Improving deep neural networks for LVCSR using rectified linear units and dropout. in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on* 8609–8613 (IEEE, 2013).

103. Erhan, D., Bengio, Y., Courville, A. & Vincent, P. Visualizing higher-layer features of a deep network. *Univ. Montr.* **1341**, 1 (2009).

104. Le, Q. V. Building high-level features using large scale unsupervised learning. in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on* 8595–8598 (2013).

105. Grün, F., Rupprecht, C., Navab, N. & Tombari, F. A Taxonomy and Library for Visualizing Learned Features in Convolutional Neural Networks. **48**, (2016).

106. Simonyan, K., Vedaldi, A. & Zisserman, A. Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps. 1–8 (2013). doi:10.1080/00994480.2000.10748487

107. Dauphin, Y. N., De Vries, H., Chung, J. & Bengio, Y. RMSProp and equilibrated adaptive learning rates for non-convex optimization (2015). arXiv preprint. *arXiv Prepr. arXiv1502.04390*

108. Bengio, Y. Deep learning of representations: Looking forward. in *International Conference on Statistical Language and Speech Processing* 1–37 (Springer, 2013).

109. Csáji, B. Approximation with artificial neural networks. *MSc. thesis* 45 (2001). doi:10.1.1.101.2647

110. Delalleau, O. & Bengio, Y. Shallow vs. deep sum-product networks. in *Advances in Neural Information Processing Systems* 666–674 (2011).

111. Wang, H. & Raj, B. On the Origin of Deep Learning. 1–72 (2017). doi:10.1016/0014-5793(91)81229-2

112. Nguyen, Q., Mukkamala, M. & Hein, M. Neural Networks Should Be Wide Enough to Learn Disconnected Decision Regions. *arXiv Prepr. arXiv1803.00094* (2018).

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

114. Lin, T.-Y. *et al.* Microsoft coco: Common objects in context. in *European conference on computer vision* 740–755 (Springer, 2014).

115. Xie, S., Girshick, R., Dollár, P., Tu, Z. & He, K. Aggregated residual transformations for deep neural networks. in *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on* 5987–5995 (IEEE, 2017).

116. Sharma, A. & Muttoo, S. K. Spatial Image Steganalysis Based on ResNeXt. *2018 IEEE 18th Int. Conf. Commun. Technol.* 1213–1216 (2018). doi:10.1109/ICCT.2018.8600132

117. Han, W., Feng, R., Wang, L. & Gao, L. 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* 4736–4739 (IEEE, 2018).

118. Dong, C., Loy, C. C., He, K. & Tang, X. Image super-resolution using deep convolutional networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **38**, 295–307 (2016).

119. Dauphin, Y. N., Fan, A., Auli, M. & Grangier, D. Language modeling with gated convolutional networks. in *Proceedings of the 34th International Conference on Machine Learning-Volume 70* 933–941 (2017).

120. Pascanu, R., Mikolov, T. & Bengio, Y. Understanding the exploding gradient problem. *CoRR, abs/1211.5063* (2012).

121. Larsson, G., Maire, M. & Shakhnarovich, G. Fractalnet: Ultra-deep neural networks

without residuals. *arXiv Prepr. arXiv1605.07648* (2016).

122. Tong, T., Li, G., Liu, X. & Gao, Q. Image super-resolution using dense skip connections. in *2017 IEEE International Conference on Computer Vision (ICCV)* 4809–4817 (2017).

123. Mao, X., Shen, C. & Yang, Y.-B. Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections. in *Advances in neural information processing systems* 2802–2810 (2016).

124. Mikolov, T., Karafiát, M., Burget, L., Černock\`y, J. & Khudanpur, S. Recurrent neural network based language model. in *Eleventh Annual Conference of the International Speech Communication Association* (2010).

125. Sundermeyer, M., Schlüter, R. & Ney, H. LSTM neural networks for language modeling. in *Thirteenth annual conference of the international speech communication association* (2012).

126. Hanin, B. & Sellke, M. Approximating Continuous Functions by ReLU Nets of Minimal Width. *arXiv Prepr. arXiv1710.11278* (2017).

127. Lu, Z., Pu, H., Wang, F., Hu, Z. & Wang, L. The expressive power of neural networks: A view from the width. in *Advances in Neural Information Processing Systems* 6231–6239 (2017).

128. Wang, Y., Wang, L., Wang, H. & Li, P. End-to-End Image Super-Resolution via Deep and Shallow Convolutional Networks. 1–10 (2016).

129. Chollet, F. Xception: Deep learning with depthwise separable convolutions. *arXiv Prepr.* 1610–2357 (2017).

130. Nixon, M. & Aguado, A. S. *Feature extraction and image processing for computer vision*. (Academic Press, 2012).

131. Hu, J., Shen, L. & Sun, G. Squeeze-and-Excitation Networks. (2017).

132. Zhou, B., Khosla, A., Lapedriza, A., Oliva, A. & Torralba, A. Learning deep features for discriminative localization. in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* 2921–2929 (2016).

133. Zheng, H., Fu, J., Mei, T. & Luo, J. Learning multi-attention convolutional neural network

for fine-grained image recognition. in *2017 IEEE International Conference on Computer Vision (ICCV)* 5219–5227 (2017).

134. Hu, Y. *et al.* Competitive Inner-Imaging Squeeze and Excitation for Residual Network. (2018). doi:arXiv:1807.08920v3

135. Dollár, P., Tu, Z., Perona, P. & Belongie, S. Integral channel features. (2009).

136. Lowe, D. G. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vis.* **60**, 91–110 (2004).

137. Minh N.Do, M. V. The Contourlet Transform: An Efficient Directional Multiresolution Image Representation. *IEEE Trans. Image Process.* **14**, 2091–2106 (2016).

138. Brandt, J. New Jersey awards $3 million for energy storage innovations. *Fierce Energy* 1 (2015). doi:10.1109/CVPR.2014.222

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

140. Vincent, P., Larochelle, H., Bengio, Y. & Manzagol, P.-A. Extracting and composing robust features with denoising autoencoders. in *Proceedings of the 25th international conference on Machine learning* 1096–1103 (ACM, 2008).

141. Hamel, P. & Eck, D. Learning Features from Music Audio with Deep Belief Networks. in *ISMIR* **10**, 339–344 (Utrecht, The Netherlands, 2010).

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

143. Goh, H., Thome, N., Cord, M. & Lim, J.-H. Top-down regularization of deep belief networks. *Adv. Neural Inf. Process. Syst.* 1–9 (2013).

144. Xu, K. *et al.* Show, attend and tell: Neural image caption generation with visual attention. in *International conference on machine learning* 2048–2057 (2015).

145. Jaderberg, M., Simonyan, K., Zisserman, A. & Kavukcuoglu, K. Spatial Transformer Networks. 1–15 (2015). doi:10.1038/nbt.3343

146. Jday, R. Caractérisation microstructurale du graphite sphéroïdal formé lors de la

solidification et à l’état solide. 946–956 (2017).

147. Kalchbrenner, N., Grefenstette, E. & Blunsom, P. A convolutional neural network for modelling sentences. *arXiv Prepr. arXiv1404.2188* (2014).

148. Collobert, R. & Weston, J. A Unified Architecture for Natural Language Processing: Deep Neural Networks with Multitask Learning. *Proc. 25th Int. Conf. Mach. Learn.* **18**, 769–783 (2008).

149. Hu, B., Lu, Z., Li, H. & Chen, Q. Topic modeling for named entity queries. in *Proceedings of the 20th ACM international conference on Information and knowledge management - CIKM ’11* **116**, 2009 (ACM Press, 2011).

150. Farfade, S. S. Farfade_14_FD_CNN_Multiview. 643–650

151. Zhang, K., Zhang, Z., Li, Z., Letters, Y. Q.-I. S. P. & 2016, undefined. Joint face detection and alignment using multitask cascaded convolutional networks. *Ieeexplore.Ieee.Org* **23**, 1499–1503 (2016).

152. Yang, S., Luo, P., Loy, C.-C. & Tang, X. From facial parts responses to face detection: A deep learning approach. in *Proceedings of the IEEE International Conference on Computer Vision* 3676–3684 (2015).

153. Ranjan, R., Patel, V. M. & Chellappa, R. A deep pyramid deformable part model for face detection. *arXiv Prepr. arXiv1508.04389* (2015).

154. Li, H., Lin, Z., Shen, X., Brandt, J. & Hua, G. A convolutional neural network cascade for face detection. in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* 5325–5334 (2015).

155. Sijin, L., Zhi-Qiang, L. & Chan, A. B. Heterogeneous Multi-task Learning for Human Pose Estimation with Deep Convolutional Neural Network. *Comput. Vis. Pattern Recognit. Work. (CVPRW), 2014 IEEE Conf.* 488–495 (2014). doi:10.1109/CVPRW.2014.78

156. Bulat, A. & Tzimiropoulos, G. Human Pose Estimation via Convolutional Part Heatmap Regression BT - Computer Vision – ECCV 2016. in (eds. Leibe, B., Matas, J., Sebe, N. & Welling, M.) 717–732 (Springer International Publishing, 2016).

157. Wang, X., Gao, L., Song, J. & Shen, H. T. Beyond Frame-level {CNN:} Saliency-Aware 3-D {CNN} With {LSTM} for Video Action Recognition. *{IEEE} Signal Process. Lett.* **24**, 510–514 (2017).

158. Wang, H. & Schmid, C. Action recognition with improved trajectories. in *Proceedings of the IEEE international conference on computer vision* 3551–3558 (2013).

159. Tran, D., Bourdev, L., Fergus, R., Torresani, L. & Paluri, M. Learning spatiotemporal features with 3d convolutional networks. in *Proceedings of the IEEE international conference on computer vision* 4489–4497 (2015).

160. Sun, L., Jia, K., Yeung, D.-Y. & Shi, B. E. Human action recognition using factorized spatio-temporal convolutional networks. in *Proceedings of the IEEE International Conference on Computer Vision* 4597–4605 (2015).

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

162. Donahue, J. *et al.* Long-term recurrent convolutional networks for visual recognition and description. in *Proceedings of the IEEE conference on computer vision and pattern recognition* 2625–2634 (2015).

163. Ji, S., Yang, M., Yu, K. & Xu, W. 3D convolutional neural networks for human action recognition. *ICML, Int. Conf. Mach. Learn.* **35**, 221–31 (2010).

164. Ren, S., He, K., Girshick, R. & Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. 1–9 (2015). doi:10.1109/TPAMI.2016.2577031

165. Dai, J., Li, Y., He, K. & Sun, J. R-FCN: Object Detection via Region-based Fully Convolutional Networks. (2016). doi:10.1016/j.jpowsour.2007.02.075

166. Gidaris, S. & Komodakis, N. Object detection via a multi-region and semantic segmentation-aware U model. *Proc. IEEE Int. Conf. Comput. Vis.* **2015 Inter**, 1134–1142 (2015).

167. Sermanet, P., Chintala, S. & LeCun, Y. Convolutional Neural Networks Applied to House Numbers Digit Classification. *Proc. 21st Int. Conf. Pattern Recognit.* 3288–3291 (2012). doi:10.0/Linux-x86_64

168. Levi, G. & Hassner, T. Sicherheit und Medien. *Sicherheit und Medien* (2009). doi:10.1109/CVPRW.2015.7301352

169. Long, Z. M., Guo, S. Q., Chen, G. J. & Yin, B. L. Modeling and simulation for the articulated robotic arm test system of the combination drive. *2011 Int. Conf. Mechatronics Mater. Eng. ICMME 2011* **151**, 480–483 (2012).

170. Cire\csan, D. C., Giusti, A., Gambardella, L. M. & Schmidhuber, J. Mitosis detection in breast cancer histology images with deep neural networks. in *International Conference on Medical Image Computing and Computer-assisted Intervention* 411–418 (2013).

171. Spanhol, F. A., Oliveira, L. S., Petitjean, C. & Heutte, L. Breast cancer histopathological image classification using Convolutional Neural Networks. in *2016 International Joint Conference on Neural Networks (IJCNN)* **29**, 2560–2567 (IEEE, 2016).

172. Spanhol, F. A., Oliveira, L. S., Petitjean, C. & Heutte, L. A dataset for breast cancer histopathological image classification. *IEEE Trans. Biomed. Eng.* **63**, 1455–1462 (2016).

173. Wahab, N., Khan, A. & Lee, Y. S. Two-phase deep convolutional neural network for reducing class skewness in histopathological images based breast cancer detection. *Comput. Biol. Med.* **85**, 86–97 (2017).

174. Abdel-Hamid, O., Mohamed, A. R., Jiang, H. & Penn, G. Applying convolutional neural networks concepts to hybrid NN-HMM model for speech recognition. *ICASSP, IEEE Int. Conf. Acoust. Speech Signal Process. - Proc.* 4277–4280 (2012). doi:10.1007/978-3-319-96145-3_2

175. Mohamed, A., Dahl, G. E. & Hinton, G. Acoustic modeling using deep belief networks. *IEEE Trans. Audio, Speech Lang. Process.* **20**, 14–22 (2012).

176. Dahl, G., Mohamed, A. & Hinton, G. E. Phone recognition with the mean-covariance restricted Boltzmann machine. in *Advances in neural information processing systems* 469–477 (2010).

177. Abdel-Hamid, O., Deng, L. & Yu, D. Exploring convolutional neural network structures and optimization techniques for speech recognition. in *Interspeech* **2013**, 1173–1175 (2013).

178. Hinton, G. E., Sabour, S. & Frosst, N. Matrix capsules with EM routing. (2018).

179. de Vries, H., Memisevic, R. & Courville, A. Deep learning vector quantization. in *European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning* (2016).

180. Szegedy, C. *et al.* Intriguing properties of neural networks.

181. Lu, H. *et al.* Wound intensity correction and segmentation with convolutional neural networks. *Concurr. Comput. Pract. Exp.* **29**, e3927 (2017).

182. Hinton, G. *et al.* Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Process. Mag.* **29**, 82–97 (2012).

183. Hinton, G. E., Krizhevsky, A. & Wang, S. D. Transforming auto-encoders. in *International Conference on Artificial Neural Networks* 44–51 (Springer, 2011).

184. Marmanis, D. *et al.* Semantic segmentation of aerial images with an ensemble of CNNs. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.* **3**, 473 (2016).

185. Kahng, M., Thorat, N., Chau, D. H. P., Viégas, F. B. & Wattenberg, M. GAN Lab: Understanding Complex Deep Generative Models using Interactive Visual Experimentation. *IEEE Trans. Vis. Comput. Graph.* **25**, 310–320 (2019).

186. Khan, A., Sohail, A. & Ali, A. A New Channel Boosted Convolutional Neural Network using Transfer Learning. *arXiv preprint arXiv:1804.08528* (2018).

187. Bhunia, A. K. *et al.* Script identification in natural scene image and video frames using an attention based Convolutional-LSTM network. *Pattern Recognit.* **85**, 172–184 (2019).

188. Geng, X. *et al.* Hardware-aware Exponential Approximation for Deep Neural Network. (2018).

189. Moons, B. & Verhelst, M. An energy-efficient precision-scalable ConvNet processor in 40-nm CMOS. *IEEE J. Solid-State Circuits* **52**, 903–914 (2017).

190. Geng, X. *et al.* Hardware-aware Softmax Approximation for Deep Neural Networks. (2018).

191. Suganuma, M., Shirakawa, S. & Nagao, T. A genetic programming approach to designing convolutional neural network architectures. in *Proceedings of the Genetic and Evolutionary Computation Conference* 497–504 (ACM, 2017).

192. Young, S. R., Rose, D. C., Karnowski, T. P., Lim, S.-H. & Patton, R. M. Optimizing deep learning hyper-parameters through an evolutionary algorithm. in *Proceedings of the Workshop on Machine Learning in High-Performance Computing Environments* 4 (ACM, 2015).

193. Khan, A., Qureshi, A. S., Hussain, M., Hamza, M. Y. & others. A Recent Survey on the Applications of Genetic Programming in Image Processing. *arXiv Prepr. arXiv1901.07387* (2019).

194. Huang, Y. *et al.* GPipe: Efficient Training of Giant Neural Networks using Pipeline Parallelism. **2014**, (2018).

195. Zhang, X., Li, Z., Loy, C. C. & Lin, D. PolyNet: A pursuit of structural diversity in very deep networks. *Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017* **2017–Janua**, 3900–3908 (2017).